UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

MULTI-OUTPUT GAUSSIAN PROCESS TOOLKIT WITH SPARSE FORMULATION FOR SPECTRAL KERNELS

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

ALEJANDRO ANDRÉS CUEVAS ACUÑA

PROFESOR GUÍA:
FELIPE TOBAR HENRÍQUEZ

MIEMBROS DE LA COMISIÓN:
JORGE SILVA SÁNCHEZ
DANIEL REMENIK ZISIS
GONZALO MENA CARRASCO

Este trabajo ha sido parcialmente financiado por los proyectos CMM Fondecyt-Iniciación 11171165 y CMM ANID PIA AFB170001

SANTIAGO DE CHILE
2020

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS
RESUMEN DE LA MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO
POR: ALEJANDRO ANDRÉS CUEVAS ACUÑA
FECHA: 2020
PROF. GUÍA: FELIPE TOBAR HENRÍQUEZ

MULTI-OUTPUT GAUSSIAN PROCESS TOOLKIT WITH SPARSE FORMULATION FOR SPECTRAL KERNELS

En modelos Bayesianos no paramétricos, los procesos Gaussianos (GP) [1] son un pilar en los problemas de regresión, los cuales se benefician de la estadística Bayesiana, mostrando propiedades atractivas, tales como ser un prior conjugado para la verosimilitud Gaussiana. Su extensión multivariada, los procesos Gaussianos con múltiples canales o salidas (MOGP) [2], apunta a incorporar información entre canales, tanto para inferencia como para predicción, para fenómenos multivariados acoplados.

Ambos tipos de GP, unicanal como multicanal están completamente determinados por su función de covarianza o kernel, que en el caso de MOGP es una función a valores matriciales. En este contexto, el mayor desafío al diseñar funciones de covarianza emerge en el balance entre generalización, es decir, el considerar una familia amplia de kernels, y al mismo tiempo mantener la condición de ser simétrico y definido positivo. una práctica común en el diseño de kernels se basa en combinarlos entre ellos, empleando operaciones como suma, producto o composición. Otra alternativa es el diseño de kernels a través de su representación espectral, siendo ejemplos de esto el kernel *Spectral Mixture* (SM) [3] para el caso unicanal, y el recientemente propuesto kernel *Multi-Output Spectral Mixture* (MOSM) [4] para multicanal.

El propósito principal de esta tesis es revisitar y extender el modelo MOSM, considerando sus principales desventajas para que este pueda ser aplicado como un modelo multicanal de propósito general. En este contexto, las principales contribuciones de este trabajo son las siguientes: primero, abordamos la indeseada escalabilidad del modelo para conjuntos de datos grandes, empleando aproximaciones *sparse* conocidas, proponiendo variables inducivas que aprovechan de mejor forma la estructura del kernel. Segundo, mejoramos en entrenamiento mediante el uso de heurísticas basadas en los datos disponibles, para encontrar puntos iniciales que beneficien el proceso de optimización. Tercero, para los casos en que los canales no están correlacionados, en los cuales puede ocurrir una transferencia negativa de información, la cual empeora la predicción, proponemos una version restringida del kernel MOSM (R-MOSM), y la complementamos usando priors regularizadores en los pesos de cada componente del kernel, ayudando a mitigar la transferencia negativa de información. Por último, la cuarta contribución de esta tesis consiste en un kit de herramientas de código abierto para MOGP, que incluye MOSM y las extensiones propuestas, además de otros kernels usados previamente en la literatura. Este kit de herramientas es llamado *Multi-Output Gaussian Process Toolkit* (MOGPTK), y fue escrito en Python, usando como base TensorFlow y GPflow.

Estas contribuciones son validadas experimentalmente usando MOGPTK [5], en conjuntos de datos multicanal de finanzas, robótica y clima.

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS
RESUMEN DE LA MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO
POR: ALEJANDRO ANDRÉS CUEVAS ACUÑA
FECHA: 2020
PROF. GUÍA: FELIPE TOBAR HENRÍQUEZ

MULTI-OUTPUT GAUSSIAN PROCESS TOOLKIT WITH SPARSE FORMULATION FOR SPECTRAL KERNELS

In Bayesian non-parametrics, Gaussian processes (GP) [1] are a staple in the regression problem which, benefiting from the Bayesian machinery, features appealing properties such as being a conjugate prior to the Gaussian likelihood. Its multivariate extension, the Multi-Output Gaussian Process (MOGP) [2], aims to incorporate information across outputs, both for inference and prediction, for coupled multivariate phenomena.

Both single-output and multi-output GP are entirely determined by the covariance function or kernel, which in the case of MOGP is a matrix-valued function. The main challenge when designing covariance functions stems from the trade-off between generality, i.e., catering for a broad class of kernels, while maintaining the positive-definiteness condition of the symmetric covariance. A common practice to kernel design is to combine kernels though, e.g., sums, products or composition. Another alternative to kernel design is though their spectral representation, as done by the Spectral Mixture (SM) kernel [3] for single output, and the recently proposed Multi-Output Spectral Mixture (MOSM) [4].

The main purpose of this thesis is to revisit and extend the MOSM model, addressing its main drawbacks so that it can be applied as a general-purpose multichannel model. In this context, the contributions of this thesis are four-fold: first, we address the poor scalability of the model for large datasets by employing known sparse approximations, and propose a set of inducing variables which better use the structure of the kernel. Second, we improve model training by defining data-driven heuristics for the initial point in the optimization. Third, for the cases where the channels are not correlated, and thus *negative transfer of knowledge* can worsen predictions, we propose a restricted version of the MOSM (R-MOSM) kernel and complement it with a regularising priors on the component weights to help mitigate the negative transfer. Lastly, the fourth contribution of this thesis is an open-source toolbox for MOGPs, which includes MOSM, other kernels, and the proposed extensions. This toolbox is called the Multi-Output Gaussian Process Toolkit (MOGPTK) and is written in Python with a TensorFlow/GPflow backend.

These contributions are validated experimentally using MOGPTK [5], on multichannel datasets from finance, robotics, climate.

Agradecimientos

Quiero agradecer al profesor Felipe Tobar, por guiarme no solo en la tesis, sino también por guiarme en gran parte del proceso de la carrera, y enseñarme cosas más allá de solo lo académico. A Taco de Wolff, por el trabajo en conjunto y porque sin él parte de esta tesis no sería posible.

A mis compañeros de eléctrica, Bárbara, Alfonso, Matías y Fernanda, por acompañarme en el paso por eléctrica, por el apoyo y compañerismo en todos los ramos que tomamos juntos, y por hacer más ameno el día a día, aunque tuviesemos que ir a la facultad un día sábado. Agradecer a la gente que me acompañó en el magister, a Camilo, Bryan, Rodrigo y Nicolas, por el apoyo y las interesantes discusiones, por esas conversaciones en los almuerzos y los cafés de la tarde. A Elsa, Taco, Antoine y Hagop, por esos almuerzos y conversaciones al café, por los consejos y la compañia en el transcurso de la tesis, por esas salidas a bares, y por mostrarme que existe un mundo por descubrir fuera del país.

Y por ultimo agradecer a mi familia, mi madre Janira, mi padre Juan y mi hermano Vicente, por el apoyo y cariño durante toda mi vida, en especial durante la carrera.

Contents

Introduction **1**
Publications . . . 3
Contributions . . . 3

1 Background **4**
1.1 Gaussian Processes . . . 5
1.2 Spectral Representation of Covariance Kernels . . . 6
1.3 Sparse Approximations . . . 7
1.3.1 Variational Inference for GP . . . 8
1.4 Multi-Output Gaussian Processes . . . 10
1.5 Existing work on MOGP . . . 12
1.5.1 Separable and Non-separable Kernels . . . 12
1.5.2 Linear Model of Corregionalization . . . 13
1.5.3 Intrinsic Corregionalization Model . . . 14
1.5.4 Semiparametric latent factor model . . . 14
1.5.5 Convolution Model . . . 15
1.5.6 Cross-Spectral Mixture . . . 16

2 The Multi-Output Spectral Mixture Kernel **17**
2.1 MOSM in a Nutshell . . . 18
2.2 Squared Exponential Spectral Mixture . . . 19
2.3 Relationship with other models . . . 20
2.4 Alternative Construction: Convolution Process . . . 21

3 Extending the MOSM Kernel **24**
3.1 Alternatives for Sparse Approximations . . . 24
3.2 Optimisation Considerations . . . 26
3.2.1 Initializing SM Kernel . . . 27
3.2.2 Initializing MOSM Kernel . . . 29
3.3 Mitigating Negative Transfer of Knowledge . . . 30
3.3.1 Restricted-MOSM (R-MOSM) . . . 31

4 Validation **33**
4.1 Experiment setting . . . 33
4.2 MOGPTK: Multi-output GP Toolkit . . . 34
4.3 Comparing initialisations . . . 35
4.4 Synthetic example of negative transfer . . . 37

4.5 Robot Inverse Dynamic Problem 40
4.6 Finance time series applications 44
4.6.1 Gold, Oil, NASDAQ, and USD index 44
4.6.2 Exchange Rates 46

Conclusions **48**

Bibliography **51**

Appendix **56**

Introduction

The regression problem is a cornerstone in machine learning, where the aim is to estimate the relationship between an output, denoted the *dependent* variable, and inputs, denoted the *independent* variable. This relationship is often estimated choosing a family of models from which a member is selected, members of said family can be determined by a finite set of parameters, in which case are denoted parametric models, whereas when the number of parameters is infinite or not a fixed quantity, are denoted non-parametric models. The input, often of an arbitrary dimension, can comprise a wide range of quantities, although the most commons are time and/or space, in contrast, the output is usually scalar-valued, this is referred as univariate regression, although the setting can be expanded to handle multiple outputs, where it is called multivariate regression, where in both cases the value which represents the outputs depends on the application.

By a Bayesian standpoint, an approach to the estimation problem is accomplished by defining (i) a *prior* distribution over the members of the family, which encapsulates the knowledge *before* incorporating the observed data, acting as a regulariser as well, (ii) a *likelihood* function, which when evaluated on a member, can be interpreted on how likely it is that the given model could have generated the observed data. Then, by using the Bayesian formalism, both the prior and the likelihood are used to obtain a (iii) *posterior* distribution over the family of models, this quantity can be interpreted as how the distribution over the family behaves *after* observing the data, this posterior can be used for estimation and forecasting. Within this context, Gaussian processes (GP) [1], have appealing qualities such as a closure of the posterior distribution under a Gaussian data likelihood, and have been widely used in regression, by employing a GP prior over continuous functions.

The behaviour of a GP is completely codified by a mean (usually assumed zero) and a covariance function commonly referred to as *kernel*, the elegance of the GP framework then comes from its ability to use different kernels and control the behaviour of the process, where the challenge of using GP flourish amidst designing a broad-class of kernels, whilst still maintaining the positive-definiteness of the symmetric covariance kernel. A common approach is to construct kernels utilising expert knowledge and incorporate it by operating different kernels, by summing, multiplying, or composing them, however, by exploiting the spectral (Fourier) representation of a kernel via the Bochner theorem [6], new covariance functions can be created, as designing kernels in the frequency domain is less restrictive than in the original input domain, a prime example of this for single output is the Spectral Mixture Kernel [3].

Moreover, the GP framework can be extended to handle multiple outputs, this is known as a Multi-Output Gaussian Process (MOGP) [2] which, by jointly modelling the outputs, is able the incorporate the information across-outputs, potentially improving the estimation. The kernel in this case becomes a vector-valued function, where the entries must model the covariance *and* cross-covariance among output, this further increases the challenge in designing flexible kernels, as designing cross-covariance kernels is difficult.

Previous approaches to MOGP [7, 2, 8] are based on linear combinations of latent factors, consisting in independent Gaussian processes, such approaches do not allow for flexibility in each output, as each latent factor has a unique set of parameters which is shared across all channels, and also all cross-covariances are symmetric. Alternatives employing the convolution process have been proposed [9], which allow for a unique set of parameters for the covariance of each output, but still suffers from having symmetric cross-covariances, in [10] a model with asymmetric cross-covariances is defined, but still is a linear combination of latent factors. The recently proposed Multi-Output Spectral Mixture (MOSM) [4] tackles this by designing a flexible family of multi-output kernels in the spectral domain, using the multivariate version of Bochner's theorem, Cramér's Theorem [11].

By considering the work of "Spectral Mixture Kernels for Multi-Output Gaussian Processes" (Advances in Neural Information Processing Systems, 2017 [4]), the main purpose of this thesis is to revisit and expand the previously proposed MOSM model, understanding the formulation of the kernel, identifying and addressing the main drawbacks of MOSM: (i) the scalability issues for large datasets, (ii) difficulty in training due to sensibility to initial point in the optimisation, (iii) hindering in the prediction in the presence of uncorrelated channels, and (iv) lack of available implementations of MOSM and other MOGP models. This is achieved by first examining the formulation of the kernel. Second, integrating current sparse GP approximations and proposing a type of inducing variables which better use the spectral structure of MOSM. Third, designing data-driven initialisations of the initial points previous to optimisation. Fourth, propose a regularised version of the MOSM kernel to mitigate the transfer of information when outputs are not correlated. And lastly, constructing a python toolkit which encapsulates all the previously mentioned work.

The outline of the work is the following: Chapter 1 reviews the relevant background of MOGP, starting from single-output GP, passing through MOGP, previous approaches, and sparse approximations for handling large datasets. Chapter 2 revisits the MOSM kernel, employing two different constructions. Chapter 3 addresses the drawbacks of MOSM using the mentioned contributions. Chapter 4 is dedicated to validate the proposed model considering synthetic and real-world data. Finally, the discussion and future work are in the Conclusion Chapter.

Publications

Part of the work which composes this thesis have been previously presented in:

- A.Cuevas, T de Wolff, F.Tobar, "Gaussian process imputation of multiple financial Series", in posters competition (first place) *EVIC*, IEEE, 2019.
- T. de Wolff, A.Cuevas, F.Tobar, "Gaussian process imputation of multiple financial Series", in *the Proc. of the IEEE International Conference of Acoustics, Speech and Signal Processing*, pp. 8444-8448, 2020.
- T. de Wolff, A.Cuevas, F.Tobar, "MOGPTK: The multi-output Gaussian process toolkit", in press at *Neurocomputing*, 2020. `https://arxiv.org/abs/2002.03471`.

Contributions

Furthermore, the main contributions of this thesis are the following:

1. To propose initialisation methods for spectral kernels, for both single and multi-output GP, finding initial points before the optimisation process. Detailed in section 3.2, the proposed initialisations use a available data and the spectral representation of the kernel to find initial estimate of the parameters.
2. To design inducing variables for MOSM, which can be incorporated in existing sparse approximation frameworks. Detailed in section 3.1, the proposed inducing variables arise from the latent factor construction of the kernel.
3. To propose a restricted version of MOSM (R-MOSM) kernel, detailed in section 3.3, which in conjunction with regularising priors, helps to mitigate the negative transfer of knowledge when uncorrelated channels are present.
4. To construct a python toolkit for multi-output GP, with implementations of previous approaches to MOGP, as well as the MOSM kernel, along with the proposed extensions. This is briefly discussed in section section 4.2.
5. To validate experimentally the proposed extensions of MOSM kernel, in section 4.3 the proposed initialisations are compared against existing methods using atmospheric CO2 data, in section 4.4 the R-MOSM is compared against regular MOSM in a synthetic dataset and in section 4.5 the proposed inducing variables are compared against existing sparse approximations in a robotics dataset.
6. To validate the effectiveness of MOSM kernel in real-world applications, learning the relationships among financial time series by modelling them through a MOGP. In section 4.6 the MOSM kernel is compared against previous MOGP frameworks in two finance datasets.

Chapter 1

Background

In this chapter we review Gaussian process methods for regression, describing the single output case and the extension to multiple outputs, we also review the construction of flexible covariances for single output using the Fourier representation of the covariance, alongside variational sparse approximations for handling large dataset, lastly, we review previous approaches to multi-output Gaussian process regression.

The regression problem can be seen as estimating a target function $g(x)$, from a set of —possibly noisy—observations of the function at the associated input, the goal is to define a family of models which will contain the estimator, and find the member, f, of said family whose evaluation explains the observations of the target function. This is usually done by minimising some performance criterion, such as a measure of error between the observations and the estimation. However, given that the objective is that the estimate should be able to generalise over other values beyond the observations, choosing from a broad family of models would produce multiple candidates for the estimation, so is common to add a regularisation term in the criterion to optimise, in order to incorporate structure on the estimate, such as smoothness or periodicity. This case when the model fits to the observations but is *not* able to generalise is known as *overfiting*.

From a Bayesian perspective, the regression problem is tackled by choosing a *prior* distribution on the estimate $f(x)$, then, in conjunction with the *likelihood* function for the dataset, and using the Bayes rule of probability, a *posterior* distribution can be found and used for prediction and forecasting. In this context, the prior encapsulates the knowledge of the function *previous* to seeing any observations, as well as contributing a regularising effect, preventing overfiting, the likelihood can be interpreted as how likely are the observations to be generated from the model, finally the posterior shows the distribution of the estimation *after* incorporating the observations, in conjunction with the prior. In this context, the Gaussian processes capitalize on the appealing properties of Bayesian estimation, where in the next section we formalise how the Gaussian processes form a robust, non-parametric, non-linear regression framework.

1.1 Gaussian Processes

A Gaussian process (GP) [1] is a non-parametric prior over functions of the form, $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ correspond to the input domain of the functions. Formally a Gaussian process f, is a collection of random variables, such that any finite collection is jointly Gaussian, said stochastic process is completely defined by a mean and covariance function, that is,

$$f(x) \sim \mathcal{GP}\left(m(x), k\left(x, x'\right)\right), \tag{1.1}$$

where the mean function is usually assumed equal to zero, $m(x) = 0$, the covariance function, also known as kernel, is parametrized in order to control structure the structure of the GP, that is $k(\cdot, \cdot) = k_\theta(\cdot, \cdot)$ for a set of parameters θ. Without loss of generality, from this point forward we will assume the input space to be, $\mathcal{X} = \mathbb{R}^p$, $p \in \mathbb{N}$.

Employing a GP prior defines a distribution over functions, considering observations of the form, $(\mathbf{x}, \mathbf{y}) = \{(x_n, y_n)\}_{n=1}^N$, with the points y_n corresponding to observations of the latent function f at input x_n, generally contaminated with a Gaussian noise, that is,

$$y_n = f(x_n) + \epsilon_n, \quad \epsilon_{\text{i}} \sim \mathcal{N}(0, \sigma_\epsilon^2), \tag{1.2}$$

this induces the posterior distribution $p(f|\mathbf{y})$, where given the marginalisation property of the multivariate Gaussian distribution this posterior can be in closed form. For a GP with zero mean, and Gaussian data likelihood, the posterior $p(f|\mathbf{y})$ is also a GP, where evaluating on a single point, $x_* \in \mathbb{R}^p$, results in a multivariate Gaussian with mean and covariance functions,

$$\begin{aligned} f(\mathbf{x}_*)|\mathbf{y} &\sim \mathcal{N}\left(\hat{\mu}, \hat{\Sigma}\right) \\ \hat{\mu} &= k_f(\mathbf{x}_*)^\top \left(K_{ff} + \sigma_\epsilon^2 I\right)^{-1} \mathbf{y} \\ \hat{\Sigma} &= k(\mathbf{x}_*, \mathbf{x}_*) - k_f(\mathbf{x}_*)^\top \left(K_{ff} + \sigma_\epsilon^2 I\right)^{-1} k_f(\mathbf{x}_*), \end{aligned} \tag{1.3}$$

where σ_ϵ^2 is the noise variance, the matrix $K_{ff} = k(\mathbf{x}, \mathbf{x})$ of size $N \times N$ correspond to the covariance of the observations, and $k_f(\mathbf{x}_*) = k(\mathbf{x}, \mathbf{x}_*)$ of size $N \times 1$, to the covariance between the observations and the function at the evaluation point.

Given a set of observations, training the model involves finding the set of kernel hyper-parameters θ that maximises the marginal likelihood, which is likelihood of the GP of generating the observations, integrating out the function values. In practice the negative log likelihood (NLL) is minimised, which takes the following expression of eq. (1.4),

$$p(\mathbf{y}|\mathbf{x}, \theta) = \log\left[\mathcal{N}(0, K_{ff} + \sigma_\epsilon^2 I)\right]. \tag{1.4}$$

Each evaluation requires the inversion of the matrix $K_{ff} + \sigma_\epsilon^2 I$ which has a cost $\mathcal{O}(N^3)$, this will be the main concern when dealing with a large number of observations and main motivation for developing sparse models.

The flexibility of GPs as a machine learning tool lies in the use of different covariance functions, which can add structure to the model by incorporating expert-knowledge by hard

coding effects in the kernel, such as periodicity or smoothness [1], between other effects, or by using flexible kernels which can adapt to a wide variety of data. However, in order to define valid covariance functions, the kernel must satisfy two conditions, it has to be (i) symmetric and (ii) positive-definite, a kernel $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is said to be symmetric if,

$$k(x, x') = k(x', x), \quad \forall\, x, x' \in \mathbb{R}^p, \tag{1.5}$$

whilst a kernel is said to be positive-definite if $\forall N \in \mathbb{N}$, $\forall \{a_1, \ldots, a_N\} \subset \mathbb{R}$, $\forall \mathbf{x} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^p$ then,

$$\sum_{n,n'}^{N} a_n a_{n'} k(x_n, x_{n'}) \geq 0, \tag{1.6}$$

that is, the kernel is positive-definite if, for any N points the $N \times N$ matrix given by $k(\mathbf{x}, \mathbf{x})$ is positive-semidefinite. Maintaining this condition is one of the main challenge in the design of new covariances, in the following section a result will be enunciated which can help fulfil this condition while maintaining flexibility.

1.2 Spectral Representation of Covariance Kernels

A kernel is said to be stationary if it can be written in the form $k(x, x') = k(x - x')$ where $(x - x')$ is usually denoted τ; as the name suggests, a stationary GP will be defined by a stationary covariance. In this context, stationary kernels are of special interest, where the Fourier representation of the kernel can be used to construct new covariance functions by designing the kernel in the spectral domain via the Bochner theorem [6],

Theorem 1.1 *(Bochner's Theorem): a function k on $\mathbb{R}^p$ is the covariance function of a weakly stationary random process on $\mathbb{R}^p$ if and only if it can be represented as,*

$$k(\tau) = \int_{\mathbb{R}^p} \mathrm{e}^{\iota\omega\cdot\tau} \mathrm{d}\nu(\omega), \tag{1.7}$$

where ν is a positive finite measure and ι is the imaginary unit.

In the case that the measure ν has a density, this density is known as the spectral density $S(\omega)$ of the covariance function, where a Fourier duality arises between the spectral density and the kernel, given by,

$$S(\omega) = \mathcal{F}\{k(\tau)\} = \int k(\tau)\mathrm{e}^{-\iota\omega\cdot\tau}\mathrm{d}\tau \tag{1.8}$$

$$k(\tau) = \mathcal{F}^{-1}\{S(\tau)\} = \int S(\omega)\mathrm{e}^{\iota\omega\cdot\tau}\mathrm{d}\omega, \tag{1.9}$$

where $\mathcal{F}$ denotes the Fourier transform operator. This relationship is known as the Wiener-Khintchin theorem [12] and allows to design the kernels in the spectral domain rather than the input domain, this is specially useful, as satisfying the positiveness of the spectral density is less restrictive than the positive definiteness of the covariance function.

A well-known example of a kernel designed using this theorem is the Spectral Mixture (SM) kernel [3], where the spectral density is formulated as a weighted mixture of Q Gaussian functions, with mixing weights a_q, spectral means μ_q and diagonal covariance Σ_q, that is,

$$\phi(\omega) = \sum_{q=1}^{Q} a_q \frac{1}{(2\pi)^{p/2}|\Sigma_q|^{1/2}} \exp\left[-\frac{1}{2}(\omega - \mu_q)^\top \Sigma_q (\omega - \mu_q)\right]. \tag{1.10}$$

Then, in order to obtain a real-valued kernel, the density is symmetrized by taking $S(\omega) = \frac{1}{2}[\phi(\omega) + \phi(-\omega)]$, where $S(\omega)$ will be the spectral density of the kernel. Using the fact that the Fourier transform of a Gaussian is also Gaussian, the SM kernel takes the form,

$$k_{SM}(\tau) = \sum_{q=1}^{Q} a_q \exp\left[-\frac{1}{2}\tau^\top \Sigma_q \tau\right] \cos(\mu_q^\top \tau), \tag{1.11}$$

where denoting the input dimension as p, the q^{th} component has a mean $\mu_q = [\mu_q^{(1)}, \ldots, \mu_q^{(p)}] \in \mathbb{R}^p$, a diagonal covariance $\Sigma_q = \text{diag}[\sigma_q^{(1)}, \ldots, \sigma_q^{(p)}]$, with $\sigma_q^{(\text{i})} \in \mathbb{R}_+$, $\text{i} = \{1, \ldots, p\}$ and mixing weights $a_q \in \mathbb{R}_+$. This flexible kernel is able to recover commonly used stationary covariances functions, such as squared exponential, Matérn, rational quadratic and periodic kernels [1].

1.3 Sparse Approximations

Training the kernel parameters of a GP is often prohibitive for large datasets, given that for the single output case, training cost is of order $\mathcal{O}(N^3)$ with N the number of training points, where having a dataset with a few thousand of points makes training unfeasible. This issue is amplified when using MOGP, where having different sets of training points for each channel further increases the required cost, where given M channels with N points each, the training cost is $\mathcal{O}(N^3M^3)$. This problem has been tackled utilising sparse approximations, such as the Partially independent training conditional (PITC) and Fully independent training conditional (FITC) [13], which use a reduced number of points, often denoted pseudo-inputs to approximate the full process, this pseudo-inputs can be either a subset of the training inputs or entirely new values of the process, for single output GP, considering K pseudo-inputs reduces the training cost to $\mathcal{O}(NK^2)$.

A downside of traditional sparse methods is that they do not approximate the full model, which can lead to overfit when optimising the inducing locations alongside the kernel parameters [14], to address this, variational methods [14] approximate the true process by minimizing the KL divergence between the approximated process and the full GP, preventing overfit while maintaining the reduced cost. This work have been expanded to be applied to non Gaussian likelihoods [15] and MOGP [16], moreover, recent methods allows to perform training in even larger datasets, using stochastic variational inference [17] which allow the training to be done employing mini batches.

1.3.1 Variational Inference for GP

Variational inference [18] (VI), similar to Markov Chain Monte Carlo (MCMC), is an approach for approximate inference. A prominent application is in the estimation of posterior distributions in Bayesian inference, where the target is finding the posterior distribution of the model, conditional on the observations denoted $\mathbf{y}$. Compared to MCMC relying in sampling from a proposal distribution, in VI the process is realised trough optimisation, with the main idea is to define an approximating family of variational distributions, $\mathcal{D}$, from which a member is chosen to be approximating distribution. The member is chosen within the family, $q \in \mathcal{D}$, such that minimises the Kullback-Leibler (KL) divergence between the true posterior, $p(f(x)|\mathbf{y})$ and the member q. Said member is denoted q^*, and satisfies the following condition,

$$q^* = \underset{q \in \mathcal{D}}{\operatorname{argmin}} \operatorname{KL}\left[q \| p(f(x)|y)\right]. \tag{1.12}$$

However, as is the case in approximate inference, the posterior distribution cannot be evaluated in closed form, and thus the KL divergence will not be computable, the workaround consist in finding an alternative objective function, which is equivalent to the KL divergence up to a constant, this alternative is the evidence lower bound (ELBO), which is obtained by expanding the objective KL divergence,

$$\begin{aligned} \operatorname{KL}\left[q||p(f(x)|\mathbf{y})\right] &= \mathbb{E}_q\left[\log q - \log p(f(x)|\mathbf{y})\right] \\ &= -\mathbb{E}_q\left[\log p(\mathbf{y}|f(x))p(f(x)) - \log q\right] + \log p(\mathbf{y}) \\ &\triangleq -\mathrm{ELBO}(q) + \log p(\mathbf{y}), \end{aligned} \tag{1.13}$$

$$\operatorname{KL}\left[q||p(f(x)|\mathbf{y})\right] \triangleq -\mathrm{ELBO}(q) + \log p(\mathbf{y}) \tag{1.14}$$

where the ELBO can be written as,

$$\mathrm{ELBO}(q) = \mathbb{E}_q\left[\log p(\mathbf{y}|f(x)) - \log \frac{q}{f(x)}\right]. \tag{1.15}$$

Now maximising the ELBO is equivalent to minimising the KL divergence as the evidence $p(\mathbf{y})$ is a constant. The name of the expression comes from the last row of eq. (1.13) and the fact that $\mathrm{KL}(\cdot) \geq 0$, where the ELBO lower-bounds the log evidence, $\log p(\mathbf{y}) \geq \mathrm{ELBO}$.

In the case of GP, the variational family is constructed considering a set of inducing-inputs (or pseudo-inputs), $\mathbf{z} = \{z_{\mathrm{i}}\}_{\mathrm{i}=1}^{K}$ which are generally in the same domain as the input of the original GP, although other alternatives will be shown in section 3.1. Then, variational family is constructed by selecting the vector of values, $\lambda = \{f(z_{\mathrm{i}})\}_{\mathrm{i}=1}^{K}$, assumed to be drawn from the same GP prior as the observations, located at the pseudo-inputs $\mathbf{z}$. Then, the form of the approximating distribution q is defined in the following manner: (i) using that the joint approximation of the posterior can be written as, $q(f(x), \lambda) = q(f(x)|\lambda)q(\lambda)$ and (ii) that in the optimal case the variational distribution q is the same as the posterior, which can be written jointly with the inducing variables $p(f(x), \lambda|\mathbf{y}) = p(f(x), \lambda)p(\lambda|\mathbf{y})$, with this we *chose* the approximating distribution to have the form,

$$q(f(x), \lambda) = p(\mathbf{f}(x)|\lambda)q(\lambda), \tag{1.16}$$

where the conditional prior $p(\mathbf{f}(x)|\lambda)$ follows eq. (1.3) but conditioning on λ, and $q(\lambda)$ will be a free-form variational distribution, which will encapsulate most of the flexibility of the model.

In order to obtain a tractable ELBO for GPs, we condition on the the inducing variables λ and latent function values $\mathbf{f}$ at the training locations $\mathbf{x}$. Conditioning on λ and $\mathbf{f}$, the prior distribution of the remainder process $f(x)$ can be written as,

$$p(f(x)) = p(f(x)|\mathbf{f}, \lambda)p(\mathbf{f}|\lambda)p(\lambda), \tag{1.17}$$

where all the terms can be obtained, as all are either Gaussian distributions or GP,

$$p(\lambda) = \mathcal{N}(0, k(\mathbf{z}, \mathbf{z})) \tag{1.18}$$

$$p(\mathbf{f}|\lambda) = \mathcal{N}(k(\mathbf{x}, \mathbf{z})K_{\mathbf{ff}}\lambda, k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \lambda)K_{\lambda\lambda}^{-1}k(\mathbf{x}, \lambda)^\top) \tag{1.19}$$

$$p(f(x)|\mathbf{f}, \lambda) = \mathcal{GP}\left(k_{\tilde{\mathbf{f}}}(x)^\top K_{\tilde{\mathbf{f}}\tilde{\mathbf{f}}}^{-1}\mathbf{y},\ k(x, x') - k_{\tilde{\mathbf{f}}}(x)^\top K_{\tilde{\mathbf{f}}\tilde{\mathbf{f}}}^{-1}k_{\tilde{\mathbf{f}}}(x')\right) \tag{1.20}$$

where $\tilde{\mathbf{f}} = [\mathbf{f}, \lambda]^\top$. Similarly, the approximating distribution $q(f(x))$ can be written as, $q(f(x)|\mathbf{f}, \lambda)q(\mathbf{f}|\lambda)q(\lambda)$, where using eq. (1.16) we note that $q(f(x)|\mathbf{f}, \lambda)$ and $q(\mathbf{f}|\lambda)$ are the same as $p(f(x)|\mathbf{f}, \lambda)$ and $p(\mathbf{f}|\lambda)$ from eq. (1.18), where the only differences will be in $p(\lambda)$ and $q(\lambda)$, with this, the ELBO from eq. (1.15) can be simplified, arriving at the following expression,

$$\text{ELBO}(q(f(x))) = \mathbb{E}_{p(\mathbf{f}|\lambda)q(\lambda)}[\log p(\mathbf{y}|\mathbf{f})] - \mathbb{E}_{q(\lambda)}\left[\log \frac{q(\lambda)}{p(\lambda)}\right]. \tag{1.21}$$

Furthermore, if $q(\lambda)$ is chosen to be a Gaussian distribution and the likelihood of the data $p(y|f(x))$ is Gaussian as well, with noise variance σ_n^2, then the ELBO at the optimal distribution q^*, and the optimal variational member itself can be found in closed form, then the ELBO is given by [14],

$$\text{ELBO}(q^*) = \log \mathcal{N}\left(y|0, Q_{ff} + \sigma_n^2 I\right) - \frac{1}{2\sigma_n^2}\text{Tr}(K_{ff} - Q_{ff}), \tag{1.22}$$

where $K_{ff} = k(\mathbf{x}, \mathbf{x})$ and $Q_{ff} = k(\mathbf{x}, \mathbf{z})k(\mathbf{z}, \mathbf{z})^{-1}k(\mathbf{z}, \mathbf{x})$ with the trace term act as a regularisation term by subtracting the difference between the real and approximated diagonal, then the optimal variational member q^* has the form,

$$\begin{aligned} q^*(\lambda) &= \mathcal{N}(\mu^*, \Sigma^*) \\ \Sigma^* &= K_{\lambda\lambda}\left(K_{\lambda\lambda} + \sigma_n^{-2}K_{\lambda\mathbf{f}}K_{\lambda\mathbf{f}}^T\right)^{-1}K_{\lambda\lambda} \\ \mu^* &= \sigma_n^{-2}\Sigma^* K_{\lambda\lambda}^{-1}K_{\lambda\mathbf{f}}\mathbf{y}. \end{aligned} \tag{1.23}$$

Then, the approximated model can be trained by optimising the ELBO at eq. (1.22) with respect to the inducing inputs $\mathbf{z}$. In order to evaluate the approximated model, the approximated distribution $q(f(x))$ can be obtained by integrating out the inducing variables λ

$$\begin{aligned} q(f(x)) = \int q(\lambda)q(f(x)|\lambda)\mathrm{d}\lambda &= \mathcal{GP}\left(\hat{\mu}_q(x), \hat{\Sigma}_q(x, x)\right), \\ \hat{\mu}_q &= k_\lambda(x)^\top K_{\lambda\lambda}^{-1}\mu^* \\ \hat{\Sigma}_q &= k(x, x) + k_\lambda(x)^T(K_{\lambda\lambda}^{-1}\Sigma^* K_{\lambda\lambda}^{-1} - K_{\lambda\lambda}^{-1})k_\lambda(x). \end{aligned}$$

The case where the data-likelihood is not Gaussian and the approximation distribution q is necessarily chosen to be Gaussian, the optimal member cannot be found in close form, although can be approximated using MCMC [19]. A more detailed derivation and discussion on the variational approximation can be found in [15].

1.4 Multi-Output Gaussian Processes

Traditional GPs can only be applied onto single a scalar-valued function, making it restrictive when the objective is to learn multiple tasks simultaneously, in which case the target function takes the form $f : \mathbb{R}^p \mapsto \mathbb{R}^M$. This problem of learning multiple task in unison, is known as multi-output learning [20] (also known as multi-task learning), and in the case of GP it can be tackled by jointly modelling M different processes, one for each output. This in turn involves learning multiple covariances -one for each output- and cross covariances across pairs of outputs, when referring to output functions, the term output and channel will be interchangeable.

A natural way to expand the GP framework to handle multiple outputs is trough a Multi-output Gaussian process (MOGP) [21, 22, 2], which consist of an augmented model where all the outputs are jointly modelled as a GP, and the covariance and cross covariances are governed by a multi-output kernel. Moreover, given the M output latent functions, $\{f_{\mathrm{i}}\}_{\mathrm{i}=1}^{M}$, the covariance kernel will be a matrix valued function $\mathcal{K} : \mathbb{R}^p \times \mathbb{R}^p \mapsto \mathbb{R}^{M\times M}$, where the element (i, j) of the kernel corresponds to the covariance between outputs f_{i} and f_j, following the notation for the (i, j) element, $[\mathcal{K}(x, x')]_{\mathrm{i}j} = k_{\mathrm{i}j}(x, x')$,

$$\mathrm{cov}[f_{\mathrm{i}}(x), f_j(x')] = k_{\mathrm{i}j}(x, x'), \quad \mathrm{i}, j = \{1, \dots, M\}. \tag{1.24}$$

Similar to the single channel case, a kernel $\mathcal{K} : \mathbb{R}^p \times \mathbb{R}^p \mapsto \mathbb{R}^{M\times M}$ is said to the covariance function of a M-output Gaussian process if: (i) is symmetric and (ii) positive-definite, where $\mathcal{K}$ is symmetric if,

$$\mathcal{K}(x, x') = \mathcal{K}(x', x)^\top, \quad \forall\, x, x' \in \mathbb{R}, \tag{1.25}$$

and positive-definite if, $\forall\, N \in \mathbb{N}$, $\forall\, \mathbf{x} = \{x_n\}_{n=1}^{N} \subset \mathbb{R}^p$, $\forall \left\{\{c_{\mathrm{i}n}\}_{\mathrm{i}=1}^{M}\right\}_{n=1}^{N} \subset \mathbb{R}$, then,

$$\sum_{\mathrm{i},j}^{M}\sum_{n,n'}^{N} c_{\mathrm{i}n}c_{jn'}k_{\mathrm{i}j}(x_n, x_{n'}) \geq 0, \tag{1.26}$$

this can be seen as that for any N points in the function domain, the $N \cdot M \times N \cdot M$ matrix $\mathcal{K}(\mathbf{x}, \mathbf{x})$ has to be positive-semidefinite. It was assumed that all channels have the same input dimension, p, but the condition can be generalized so each output can have different input dimension, the same can be said about the number of observations, where each channel can have a different number of observations.

Fitting the model is akin to the single output case, consisting in finding the kernel parameters Θ which minimize the negative log marginal likelihood (NLL), moreover, denoting N_{i} observations for channel i, the total observations of channel i by,

$$(X_{\mathrm{i}}, Y_{\mathrm{i}}) = \left\{(x_n^{(\mathrm{i})}, y_n^{(\mathrm{i})})\right\}_{n=1}^{N_{\mathrm{i}}}, \quad \mathrm{i} = \{1, \dots, M\}, \tag{1.27}$$

the noise variance for channel i by σ_{i}^2, $\hat{N} = \sum_{\mathrm{i}=1}^{M} N_{\mathrm{i}}$ the total number of observations, $\mathbf{Y}$ the vector of concatenated observations, $\mathbf{X}$ the matrix of concatenated inputs,

$$\begin{aligned}\mathbf{Y} &= [y_1^{(1)}, \ldots, y_{N_1}^{(1)}, \ldots, y_1^{(M)}, \ldots, y_{N_M}^{(M)}] \\ &= [Y_1, \ldots, Y_M] \\ \mathbf{X} &= [x_1^{(1)}, \ldots, x_{N_1}^{(1)}, \ldots, x_1^{(M)}, \ldots, x_{N_M}^{(M)}] \\ &= [X_1, \ldots, X_M],\end{aligned} \tag{1.28}$$

and the concatenated noise variances $\Lambda = \mathrm{di}ag[\sigma_1^2, \ldots, \sigma_m^2]$. Then, the gram matrix $\mathcal{K}(\mathbf{X}, \mathbf{X}')$ will be an $\hat{N} \times \hat{N}$ block matrix where each block of size $N_{\mathrm{i}} \times N_j$ will contain the covariance between channels i, j, between all observations on said channels, and the block diagonal contains the covariance of an output with itself. With this the NLL has the form,

$$\begin{aligned}\mathrm{NLL} &= -\log p(\mathbf{Y}|\mathbf{X}, \Theta) \\ &= \frac{\hat{N}}{2}\log 2\pi + \frac{1}{2}\log|K_{yy}| + \frac{1}{2}\mathbf{Y}^\top K_{yy}^{-1}\mathbf{Y},\end{aligned} \tag{1.29}$$

with $K_{yy} = \mathcal{K}(\mathbf{X}, \mathbf{X}') + \Lambda$. The evaluation of the model follows from the fact that any collection of values of the process are jointly Gaussian, regardless of the channel, with this, the posterior distribution for a single input point, $x_* \in \mathbb{R}^p$ is given by,

$$\begin{aligned}f(x_*)|\mathbf{Y} &\sim \mathcal{N}\left(\hat{\mu}, \hat{\Sigma}\right) \\ \hat{\mu} &= k_f(x_*)^\top K_{yy}^{-1}\mathbf{Y} \\ \hat{\Sigma} &= \mathcal{K}(x_*, x_*) - k_{\mathrm{i}f}(x_*)^\top K_{yy}^{-1} k_f(x_*),\end{aligned} \tag{1.30}$$

with $\hat{\mu} \in \mathbb{R}^M$ and $\hat{\Sigma} \in \mathbb{R}^{M\times M}$, where $k_f(x_*)$ of size $N \cdot M \times M$ is the covariance between the observations and the evaluation point, that is,

$$\begin{aligned}k_f(x_*) &= \mathcal{K}(\mathbf{X}, x_*) \\ &= \left[k_{\bullet 1}\left(\mathbf{X}, x_*\right), \ldots, k_{\bullet M}\left(\mathbf{X}, x_*\right)\right].\end{aligned} \tag{1.31}$$

This posterior distribution resembles the single output case, but now additional cross-channel information can be taking into account. The MOGP can also be evaluated for a single output, where the posterior for a single channel i evaluated at the same input x_* is given by,

$$\begin{aligned}f_{\mathrm{i}}(x_*)|\mathbf{Y} &\sim \mathcal{N}\left(\hat{\mu}_{\mathrm{i}}, \hat{\Sigma}_{\mathrm{i}}\right) \\ \hat{\mu}_{\mathrm{i}} &= k_{\mathrm{i}f}(x_*)^\top K_{yy}^{-1}\mathbf{Y} \\ \hat{\Sigma}_{\mathrm{i}} &= k_{\mathrm{ii}}(x_*, x_*) - k_{\mathrm{i}f}(x_*)^\top K_{yy}^{-1} k_{\mathrm{i}f}(x_*),\end{aligned} \tag{1.32}$$

where $\hat{\mu}_{\mathrm{i}} \in \mathbb{R}$, $\hat{\Sigma}_{\mathrm{i}} \in \mathbb{R}^{1\times 1}$ and $k_{\mathrm{i}f}(x_*) = k_{\bullet\mathrm{i}}(\mathbf{X}, x_*)$ of size $N \cdot M \times 1$.

One of the main challenges in MOGP lies in designing expressive covariances while fulfilling the positive-definiteness condition. In the next section prior work on MOGP will be highlighted and compared with the proposed model.

1.5 Existing work on MOGP

The applications of multi-output learning can be traced in a wide spectrum of fields, where a pioneer has been in the area of geostatistics, using models such as the Linear model of corregionalization [7], where the use of MOGP it is known as *cokriging*. MOGP has also been used in robotics [23] and recently in financial time series [24]. In the following section we will contextualize previous approaches to MOGP, where in the subsequent section these models will be shown to be a particular case of the proposed model.

1.5.1 Separable and Non-separable Kernels

Multi-output kernels generally can be classified in separable and non-separable [2], separable kernels are such that the multi-output covariance is composed by the multiplication of a kernel that only depend in the input variables, and a kernel that only depend of the channels. The components of the multi-output kernels take the form,

$$k_{\mathrm{i}j}(x, x') = k(x, x')\, k_a(\mathrm{i}, j), \tag{1.33}$$

where the contribution of inputs and outputs are decoupled. A common choice of the output kernel, k_a, is one that associates coefficients to each output channel, that is, $k_a(\mathrm{i}, j) = a_{\mathrm{i}} a_j$, then the multi-output kernel can be written as a base kernel multiplied by coefficients depending on the output,

$$k_{\mathrm{i}j}(x, x') = a_{\mathrm{i}} a_j\, k(x, x'). \tag{1.34}$$

Then, we can group the coefficients and write the multi-output kernel in matrix form,

$$\begin{aligned} \mathcal{K}(x, x') &= k(x, x')\, a^\top a \\ \mathcal{K}(x, x') &= k(x, x') B, \end{aligned} \tag{1.35}$$

where $a = [a_1, \ldots, a_M]$ and B is a $M \times M$ matrix, usually of real values, although we will see cases with complex-valued coefficients. A generalization on this is to consider a sum of Q kernels, k_q, yielding the expression for sum of separable kernels,

$$\mathcal{K}(x, x') = \sum_q^Q k_q(x, x') B, \tag{1.36}$$

where for a set of inputs $\mathbf{X} = \{x_n\}_{n=1}^N$ the kernel can be written as,

$$\mathcal{K}(\mathbf{X}, \mathbf{X}) = \sum_q^Q B_q \otimes k_q(\mathbf{X}, \mathbf{X}), \tag{1.37}$$

where $\otimes$ denotes the Kronecker product between matrices.

In contrast, a *non*-separable kernel is such that cannot be written in the form eq. (1.37) and thus the entry (i, j) on the kernel depends on a single function which incorporates inputs and channels,

$$k_{\mathrm{i}j}(x, x') = k(x, x', \mathrm{i}, j). \tag{1.38}$$

As it will be shown, separable kernels tend to be more simple to construct and train, given that the number of kernel parameters is reduced, only needing the base kernel parameters and output coefficients, and the fact that the positive-definiteness is given by construction if $B = a^\top a$, and k is valid single output kernel. In the non-separable case the flexibility comes to the cost of no structural benefit to ensure positive-definiteness, this proposes the challenging problem of designing flexible non-separable kernel while maintaining the positive-definiteness condition.

In the next section, previous approaches to MOGP kernels will be briefly discussed, highlighting the most common models.

1.5.2 Linear Model of Corregionalization

The linear model of corregionalization (LMC) [7], models each output as a linear combination of independent latent processes, $\{u_q\}_{q=1}^{Q}$, where this latent processes are Gaussian processes. Given that the linear combination of GPs is also a GP, each output takes the form,

$$f_{\mathrm{i}}(x) = \sum_{q=1}^{Q} a_{\mathrm{i}q} u_q(x), \quad \mathrm{i} = \{1, \dots, M\}, \tag{1.39}$$

where the latent processes u_q, have zero mean and covariance $\mathrm{cov}[u_q(x), u_{q'}(x')] = \delta_{qq'} k_q(x, x')$, with $\delta_{qq'}$, the Kronecker delta. Furthermore, some latent function can have the same kernel while remaining independent, and subsequently the latent processes that share the same covariance can be grouped together. Expressing the output functions f_{i} as,

$$f_{\mathrm{i}}(x) = \sum_{q=1}^{Q} \sum_{r=1}^{R_q} a_{\mathrm{i}q}^{(r)} u_q(x)^{(r)}, \quad \mathrm{i} = \{1, \dots, M\}, \tag{1.40}$$

where there are Q different covariances of the latent processes, with the component q having R_q independent replicas with the same covariance. With the above characterisation of the output functions, the covariance between outputs can be obtained and the multi-output kernel constructed, with this, the covariance between the outputs i and j, $\mathrm{cov}[f_{\mathrm{i}}, f_j] = k_{\mathrm{i}j}$ takes the following expression,

$$\begin{aligned} k_{\mathrm{i}j}(x, x') &= \sum_{q=1}^{Q} \sum_{r=1}^{R_q} a_{\mathrm{i}q}^{(r)} a_{jq}^{(r)} k_q(x, x') \\ &= \sum_{q=1}^{Q} b_{\mathrm{i}j}^{(q)} k_q(x, x'), \end{aligned} \tag{1.41}$$

with $b_{\mathrm{i}j}^{q} = \sum_{r=1}^{R_q} a_{\mathrm{i}q}^{(r)} a_{jq}^{(r)}$, then, we can write and expression for the covariance of the joint process, utilising all the outputs,

$$\mathcal{K}(x, x') = \sum_{q=1}^{Q} B_q k_q(x, x'), \tag{1.42}$$

where the matrix $B_q \in \mathbb{R}^{M\times M}$ is given by $[B_q]_{\mathrm{i}j} = b_{ij}^{(q)}$, this matrix is known as *corregionalization* matrix [7], the LMC kernel follows a structure of sum of separable kernels, thus the kernel in eq. (1.42) will be positive-definite as long as the corregionalization matrix is positive-semidefinite and the kernels k_q are valid kernels, where B_q is positive semidefinite by definition.

1.5.3 Intrinsic Corregionalization Model

The intrinsic Corregionalization Model (ICM) [7] is a simplification of the LMC, by assuming that elements $b_{ij}^{(q)}$ of the corregionalization matrix B_q can be written as $b_{ij}^{(q)} = v_{\mathrm{i}j}\, b_q$, that is, decouples the dependency of the outputs and the latent components, using this model, the covariance between channels takes the form,

$$k_{\mathrm{i}j}(x, x') = v_{\mathrm{i}j} \sum_{q=1}^{Q} b_q k_q(x, x'), \tag{1.43}$$

where the joint covariance can be written as,

$$\begin{aligned} \mathcal{K}(x, x') &= \Upsilon \sum_{q=1}^{Q} b_q k_q(x, x') \\ &= \Upsilon\, \hat{k}(x, x') \end{aligned} \tag{1.44}$$

where $\Upsilon \in \mathbb{R}^{M\times M}$ has entries $[\Upsilon]_{\mathrm{i}j} = v_{\mathrm{i}j}$ and $\hat{k} = \sum_{q=1}^{Q} b_q k_q$. From eq. (1.44) it can be seen as a LMC kernel with $Q = 1$, and as a particular case of LMC, will be a valid multi-output kernel. As was pointed in [7], this construction is more restrictive model than LMC, where now each component kernel k_q will contribute equally to covariance and cross-covariances between outputs.

1.5.4 Semiparametric latent factor model

The semiparametric latent factor model [25], results in a particular case of LMC obtained when considering $\{R_q = 1\}_{q=1}^{Q}$, that is,

$$k_{\mathrm{i}j}(x, x') = \sum_{q=1}^{Q} a_{\mathrm{i}q} a_{jq} k_q(x, x'), \tag{1.45}$$

which can be written for all outputs in a similar form of sum of separable kernels of eq. (1.37) and eq. (1.42),

$$\mathcal{K}(x, x') = \sum_{q=1}^{Q} B_q k_q(x, x'), \tag{1.46}$$

where now the coefficient matrix $B_q \in \mathbb{R}^{M\times M}$ will have rank 1 and the elements are given by $[B_q]_{\mathrm{i}j} = a_{\mathrm{i}q} a_{jq}$. The semiparametric latent factor model name comes from the linear (parametric) mixing of random processes, u_q, which are nonparametric GPs.

1.5.5 Convolution Model

Previous models such as LMC are based in the idea of linear combinations of latent factors, this can be seen as a *instantaneous* mixing of latent factors, as each factor $u_q(x)$, at a input x, only affects an output $f_{\mathrm{i}}(x)$, at the instant same instant x. Based on the idea of using convolutions instead of instantaneous mixing, the convolution model [9] model each output as the convolution of the latent factors with a smoothing kernel, that is, for outputs $\{f_{\mathrm{i}}\}_{\mathrm{i}=1}^{M}$,

$$\begin{aligned} f_{\mathrm{i}}(x) &= \sum_{q=1}^{Q} \int_{\mathcal{X}} h_{\mathrm{i}q}(x-z) u_q(z) \mathrm{d}z \\ &= \sum_{q=1}^{Q} h_{\mathrm{i}q}(x) \star u_q(x), \end{aligned} \tag{1.47}$$

where $\star$ denotes the convolution operator. Assuming each u_q independent of each other, that is, $\mathrm{cov}[u_q, u_{q'}] = \sigma_{uq}^2 \delta_{qq'}$, then the covariance between outputs takes the form,

$$k_{\mathrm{i}j}(x, x') = \sum_{q=1}^{Q} \int_{\mathcal{X}} \int_{\mathcal{X}} h_{\mathrm{i}q}(x-z) h_{jq}(x'-z') k_q(z, z') \mathrm{d}z' \mathrm{d}z. \tag{1.48}$$

Furthermore, if the latent functions, u_q, are assumed to be independent white noise processes, $\mathrm{cov}[u_q(z), u_{q'}(z')] = \sigma_{uq}^2 \delta_{qq'} \delta(z-z')$, consequently the eq. (1.48) is reduced to just one convolution,

$$\begin{aligned} k_{\mathrm{i}j}(x, x') &= \sum_{q=1}^{Q} \sigma_{uq}^2 \int_{\mathcal{X}} h_{\mathrm{i}q}(x-z) h_{jq}(x'-z) \mathrm{d}z \\ &= \sum_{q=1}^{Q} \sigma_{uq}^2 \int_{\mathcal{X}} h_{\mathrm{i}q}(\tilde{z}) h_{jq}(x'-x+\tilde{z}) \mathrm{d}\tilde{z}, \\ &= \sum_{q=1}^{Q} \sigma_{uq}^2 \int_{\mathcal{X}} h_{\mathrm{i}q}(\tilde{z}) h_{jq}(-(\tau-\tilde{z})) \mathrm{d}\tilde{z}, \\ &= \sum_{q=1}^{Q} \sigma_{uq}^2 \left(h_{\mathrm{i}q}(\tau) \star h_{jq}(-\tau)\right)(\tau), \end{aligned} \tag{1.49}$$

where $\tilde{z} = x - z$ and $\tau = x - x'$, this covariance function will be stationary as it only depend on the difference between the inputs τ. The convolution process correspond to a generalization of LMC, where if the smoothing kernels $h_{\mathrm{i}q}$ are taken to be Dirac delta function such that, $h_{\mathrm{i}q}(x-z) = a_{\mathrm{i}q}\delta(x-z')$ then the eq. (1.42) is recovered.

Since the framework requires the convolution to be evaluated in closed form, in [9] propose the use of Gaussian functions for smoothing kernel and the covariance of the latent process, due to the fact that Gaussian functions are closed under convolution. With this, the *Gaussian convolution model* (CONV) has a *non*-separable covariance function, where the smoothing kernels are given by $h_{\mathrm{i}q} = (2\pi)^{-p/2} |\Sigma_{\mathrm{i}q}|^{1/2} \mathrm{e}^{\left(-\frac{1}{2} x^\top \Sigma_{\mathrm{i}q} x\right)}$, and the covariance of the

latent processes, $\text{cov}[u_q, u_{q'}] = k_q$, are given by, $k_q(\tau) = (2\pi)^{-p/2}|\Sigma_q|^{1/2}\text{e}^{\left(-\frac{1}{2}(\tau)^\top\Sigma_q(\tau)\right)}$, then, the multi-output kernel takes the following expression,

$$k_{\text{i}j}(\tau) = \sum_{q=1}^{Q} a_{\text{i}q}a_{jq}\frac{|\Sigma_{\text{i}jq}|^{1/2}}{(2\pi)^{p/2}}\exp\left(-\frac{1}{2}(\tau)^\top\Sigma_{\text{i}jq}(\tau)\right), \tag{1.50}$$

where $\Sigma_{\text{i}jq}^{-1} = \Sigma_q^{-1} + \Sigma_{\text{i}q}^{-1} + \Sigma_{\text{i}q}^{-1}$. A mayor difference of CONV with separable kernels such as LMC is that as a non-separable kernel, CONV can assign each output with different kernel parameters, whereas in LMC, the kernel parameters of k_q are shared across all outputs. As was noted in [9], when input dimension, denoted p, has a high dimension, then normalisation term $(2\pi)^{p/2}$ will dominate in eq. (1.50), making the value of the kernel decay to zero, a solution to this is scale the channels standard deviation of the kernel.

1.5.6 Cross-Spectral Mixture

The cross-spectral mixture (CSM) [10] kernel is a separable kernel, which was proposed as a generalization on the LMC framework, allowing corregionalization matrices, B on eq. (1.42) to take complex values, thus, being able to obtain non-symmetric cross-covariances. The model is constructed by taking the latent factors, u_q, to be a GP with spectral mixture kernel, that is, k_q follows eq. (1.11), subsequently, the corregionalization matrices, B_q are chosen to be,

$$[B_q]_{\text{i}j} = \sum_{r=1}^{R_q}\sqrt{a_{\text{i}q}^{(r)}a_{jq}^{(r)}}\exp(-\iota(\varphi_{\text{i}q}^{(r)} - \varphi_{jq}^{(r)})), \tag{1.51}$$

then, cross spectral mixture kernel is formulated as,

$$k_{\text{i}j}(x, x') = \text{Re}\left\{\sum_{q=1}^{Q} B_q\,\tilde{k}_{SW}^{(q)}(x, x')\right\} \tag{1.52}$$

where $\tilde{k}_{SM}^{(q)} = \exp\left(-\frac{1}{2}\tau^\top\Sigma_q\tau + \iota\mu_q^\top\tau\right)$ is the complex-valued, non-symmetrised version of the SM kernel of eq. (1.11), where Σ_q is a diagonal matrix. With this, the CSM takes the form,

$$k_{\text{i}j}(x, x') = \sum_{q=1}^{Q}\sum_{r=1}^{R_q}\sqrt{a_{\text{i},q}^{(r)}a_{jq}^{(r)}}\exp\left(-\frac{1}{2}\tau^\top\Sigma_q\tau\right)\cos\left(\mu_q^\top\tau + \left(\varphi_{\text{i}q}^{(r)} - \varphi_{jq}^{(r)}\right)\right). \tag{1.53}$$

To the best of our knowledge, this kernel is the first to incorporate non-symmetric cross-covariances, in the form of the cross-phases, $\varphi_{\text{i}q}^{(r)}$, this hints the construction of flexible covariance functions for multi-output GP where additional to incorporating phase shifts between channels, other variations can be added such as time delay in the input values of the kernel.

Chapter 2

The Multi-Output Spectral Mixture Kernel

In this chapter, the formulation of the multi-output spectral mixture kernel (MOSM) [4] will be revisited, considering two different constructions, the first one consists in constructing a family of Hermitian definite-positive spectral densities, in a similar way of how the SM kernel is constructed for single output, but using the Cramér's Theorem [11] instead, with can be seen as the multi-output version of Bochner theorem. The second form of constructing the MOSM kernel employs the convolution process framework, where latent factors are convolved with filters, so that covariance of the filtered process results in the MOSM kernel.

Theorem 2.1 *(Cramér's Theorem): A family* $\{k_{\mathrm{i}j}(\tau)\}_{\mathrm{i},j=1}^{M}$ *of integrable functions are the covariance functions of a weakly-stationary multivariate stochastic process if and only if they (i) admit the representation,*

$$k_{\mathrm{i}j}(\tau) = \int_{\mathbb{R}^p} \mathrm{e}^{\iota\omega^\top\tau} S_{\mathrm{i}j}(\omega)\mathrm{d}\omega \quad \forall\, \mathrm{i}, j \in 1, \ldots, M, \tag{2.1}$$

where each $S_{\mathrm{i}j}$ *is an integrable complex-valued function* $S_{\mathrm{i}j} : \mathbb{R}^p \to \mathbb{C}$*, known as the spectral density associated to the covariance function* $k_{\mathrm{i}j}(\tau)$*, and (ii) fulfil the positive definiteness condition,*

$$\sum_{\mathrm{i},j=1}^{M} \overline{a_{\mathrm{i}}} a_j S_{\mathrm{i}j}(\omega) \geq 0 \quad \forall\, \{a_1, \ldots, a_M\} \subset \mathbb{C}, \omega \in \mathbb{R}^p. \tag{2.2}$$

Where similar to the Bochner theorem, from eq. (2.1) the cross-covariance kernels $k_{\mathrm{i}j}$ and the corresponding spectral densities $S_{\mathrm{i}j}$ are Fourier pairs, furthermore, the argument of the kernel, $\tau \in \mathbb{R}^p$ lies in the input domain of the GP, whereas the argument of the spectral densities, $\omega \in \mathbb{R}^p$, corresponds to the frequency domain. It is also worth noting from eq. (2.2), for any ω the evaluation of the spectral density, $S(\omega) = [S_{\mathrm{i}j}]_{\mathrm{i},j=1}^{M}$ yields a positive-definite $M \times M$ matrix, and thus, said matrix will be hermitian, that is, for any ω, $S(\omega) = \overline{S(\omega)^\top}$.

This theorem enables the construction of covariance functions in the spectral domain rather than the input domain, where a key aspect is that the conditions to be fulfilled for

the spectral densities in eq. (2.2) are less restrictive than the needed for the covariance in eq. (1.26), this can be better understood when comparing to the single output case, where positive-definiteness in the kernel only requires positiveness in the spectral density, whilst in the multi-output case the positive-definiteness of the kernel only requires positive-definiteness of the matrix $S(\omega)$, for all ω, this without imposing any restriction on the components $S_{\mathrm{i}j}(\omega)$. Another way interpretation is that for the kernel, for any N input points, $\mathbf{x} = \{x_n\}_{n=1}^{N}$, the matrix given by $K(\mathbf{x}, \mathbf{x})$ of size $N \cdot M \times N \cdot M$ has to be positive-semidefinite whereas for the spectral density, for any ω the matrix $S(\omega)$ of size $M \times M$ needs to be positive-semidefinite.

2.1 MOSM in a Nutshell

Recall that the covariance kernel of a MOGP of M channels can be thought of as family of covariance functions,

$$\{k_{\mathrm{i}j}(\tau) : \mathbb{R}^p \mapsto \mathbb{R}\}, \tag{2.3}$$

such that when used in conjunction as a $M \times M$ matrix-valued function, is symmetric, eq. (1.25), and positive-definite, eq. (1.26). We now describe a procedure to generate this family, starting from an arbitrary family of functions and perform a series of transformations such that the resulting family fulfils the conditions of a MOGP kernel, said arbitrary family of functions will be denoted by $\mathcal{R} = \{\hat{r}_{\mathrm{i}}(\cdot)\}_{\mathrm{i}=1}^{M}$.

Proposition 2.2 *Let us consider an arbitrary $\mathbb{C}^{Q\times M}$-valued function*

$$\hat{h}(\omega) : \Omega \mapsto \mathbb{C}^{Q\times M}, \tag{2.4}$$

with $\Omega \subseteq \mathbb{R}^p$, such that each component of $\hat{h}(\omega)$ lies in L^1 and is bounded, and denote by $\hat{k}(\omega)$ the outer product of $\hat{h}(\omega)$ with itself, that is,

$$\begin{aligned} \hat{k} : \Omega &\mapsto \mathbb{C}^{M\times M} \\ \omega &\to \hat{k}(\omega) = \hat{h}^{\mathsf{H}}(\omega)\hat{h}(\omega). \end{aligned} \tag{2.5}$$

Then, the inverse Fourier transform of $\hat{k}$, given by,

$$k(\tau) = \mathcal{F}^{-1}\left\{\hat{k}(\omega)\right\}, \tag{2.6}$$

is a valid stationary covariance kernel of an M-channel stochastic process.

Proof. To show that $k(\tau)$ is a valid covariance function, it needs to fulfil the two conditions of Cramér's theorem, the first condition comes from construction in eq. (2.6), and the fact that the Fourier transform to a matrix-valued function is component by component, thus the spectral density associated to $k_{\mathrm{i}j}$ is $\hat{k}_{\mathrm{i}j}$, note that the inverse Fourier transform is well defined, as product of bounded Lebesgue integrable functions are Lebesgue integrable. The second condition is met as $\hat{k}_{\mathrm{i}j}$ is Hermitian and positive-definite by construction from eq. (2.5). □

The strength of the above Proposition is that we could consider *any* $Q \cdot M$ functions (provided that they are bounded and Lebesgue integrable) and the above procedure will

deliver a family of valid stationary covariance functions, where taking taking a complex-vector-valued function of size $Q \times M$, a covariance is obtained after computing the inner product with itself and then applying the inverse Fourier transform.

2.2 Squared Exponential Spectral Mixture

Given this generative framework, we can construct a family of Hermitian positive-definite complex-valued functions and use them as cross-spectral densities whiting MOGP. Since Fourier transform and multiplication of squared exponential (SE) functions are also SE, we propose $R(\omega) = \hat{h}(\omega)$ to be a complex valued SE function, where $R \in \mathbb{C}^{Q \times m}$. For the sake of simplicity we chose $Q = 1$, where case for arbitrary Q is shown in the end of the section, denoting R_{i} the i^{th} component (in the case of arbitrary Q the i^{th} column) of R,

$$R_{\mathrm{i}}(\omega) = a_{\mathrm{i}} \exp\left(-\frac{1}{4}(\omega - \mu_{\mathrm{i}})^{\top}\Sigma_{\mathrm{i}}^{-1}(\omega - \mu_{\mathrm{i}})\right) \exp\left(-\iota(\theta_{\mathrm{i}}^{\top}\omega + \phi_{\mathrm{i}})\right), \quad \mathrm{i} = 1, \ldots, M, \tag{2.7}$$

subsequently, the power spectral density will be $S_{\mathrm{i}j}(\omega) = R_{\mathrm{i}}^{\mathsf{H}} R_j$, that is,

$$S_{\mathrm{i}j}(\omega) = a_{\mathrm{i}j} \exp\left(-\frac{1}{2}(\omega - \mu_{\mathrm{i}j})^{\top}\Sigma_{\mathrm{i}j}^{-1}(\omega - \mu_{\mathrm{i}j}) + \iota(\theta_{\mathrm{i}j}^{\top}\omega + \phi_{\mathrm{i}j})\right), \quad \mathrm{i}, j = 1, \ldots, M. \tag{2.8}$$

Note that this is a Gaussian, with spectral mean $\mu_{\mathrm{i}j}$ and covariance $\Sigma_{\mathrm{i}j}$, multiplied with a complex exponential, containing the information on the delay $\theta_{\mathrm{i}j}$ and phase $\phi_{\mathrm{i}j}$. The cross channel parameters (denoted by subscripts $\mathrm{i}j$) follow the next relations and interpretations,

- Magnitude: $a_{\mathrm{i}j} = a_{\mathrm{i}} a_j \exp\left(-\frac{1}{4}(\mu_{\mathrm{i}} - \mu_j)^{\top}(\Sigma_{\mathrm{i}} + \Sigma_j)^{-1}(\mu_{\mathrm{i}} - \mu_j)\right)$
- Covariance: $\Sigma_{\mathrm{i}j} = 2(\Sigma_{\mathrm{i}}^{-1} + \Sigma_j^{-1})^{-1} = 2\Sigma_{\mathrm{i}}(\Sigma_{\mathrm{i}} + \Sigma_j)^{-1}\Sigma_j$
- Mean: $\mu_{\mathrm{i}j} = (\Sigma_{\mathrm{i}}^{-1} + \Sigma_j^{-1})^{-1}(\Sigma_j^{-1}\mu_j + \Sigma_{\mathrm{i}}^{-1}\mu_{\mathrm{i}}) = (\Sigma_{\mathrm{i}} + \Sigma_j)^{-1}(\Sigma_{\mathrm{i}}\mu_j + \Sigma_j\mu_{\mathrm{i}})$
- Delay: $\theta_{\mathrm{i}j} = \theta_{\mathrm{i}} - \theta_j$
- Phase: $\phi_{\mathrm{i}j} = \phi_{\mathrm{i}} - \phi_j$

In order to restrict this generative model to real-valued GPs, the power spectral density needs to be real and symmetric with respect to w, we make $S_{\mathrm{i}j}(\omega)$ symmetric by reassigning

$$S_{\mathrm{i}j}(\omega) \leftarrow \frac{1}{2}\left(S_{\mathrm{i}j}(\omega) + S_{\mathrm{i}j}(-\omega)\right), \tag{2.9}$$

this ensures symmetry with respect to ω, and real values when taking $\mathrm{i} = j$ as the complex terms in eq. (2.10) cancel each other,

$$S_{\mathrm{i}j}(\omega) = \frac{a_{\mathrm{i}j}}{2}\left(\mathrm{e}^{\left(-\frac{1}{2}(\omega-\mu_{\mathrm{i}j})^{\top}\Sigma_{\mathrm{i}j}^{-1}(\omega-\mu_{\mathrm{i}j})+\iota(\theta_{\mathrm{i}j}^{\top}\omega+\phi_{\mathrm{i}j})\right)} + \mathrm{e}^{\left(-\frac{1}{2}(\omega+\mu_{\mathrm{i}j})^{\top}\Sigma_{\mathrm{i}j}^{-1}(\omega+\mu_{\mathrm{i}j})+\iota(-\theta_{\mathrm{i}j}^{\top}\omega+\phi_{\mathrm{i}j})\right)}\right). \tag{2.10}$$

Note that this construction makes the off-diagonal elements ($\mathrm{i} \neq j$) able to take complex-values and be asymmetric w.r.t $\omega = 0$, while still keeping the diagonal ($\mathrm{i} = j$) real-valued and symmetric, thus fulfilling the conditions of a power spectral density, this allow for flexible

modelling the cross-covariances. Finally, the kernel is obtained by taking the inverse Fourier transform of the symmetrised spectral density,

$$
\begin{aligned}
k_{\mathrm{i}j}(\tau) &= \mathcal{F}^{-1}\left\{S_{\mathrm{i}j}(\omega)\right\} \\
k_{\mathrm{i}j}(\tau) &= \alpha_{\mathrm{i},j} \exp\left(-\frac{1}{2}(\tau+\theta_{\mathrm{i}j})^{\top}\Sigma_{\mathrm{i}j}(\tau+\theta_{\mathrm{i}j})\right)\cos\left((\tau+\theta_{\mathrm{i}j})^{\top}\mu_{\mathrm{i}j}+\phi_{\mathrm{i}j}\right),
\end{aligned} \tag{2.11}
$$

where $\alpha_{\mathrm{i}j} = a_{\mathrm{i}j}(2\pi)^{\frac{n}{2}}|\Sigma_{\mathrm{i}j}|^{1/2}$. This kernel can be expanded to higher rank matrix by taking $Q > 1$, arriving at the proposed kernel,

Definition 2.3 *The Multi-Output Spectral Mixture Kernel (MOSM) has the form:*

$$
\boxed{k_{\mathrm{i}j}(\tau) = \sum_{q=1}^{Q} \alpha_{\mathrm{i}j}^{(q)} \exp\left(-\frac{1}{2}(\tau+\theta_{\mathrm{i}j}^{(q)})^{\top}\Sigma_{\mathrm{i}j}^{(q)}(\tau+\theta_{\mathrm{i}j}^{(q)})\right)\cos\left((\tau+\theta_{\mathrm{i}j}^{(q)})^{\top}\mu_{\mathrm{i}j}^{(q)}+\phi_{\mathrm{i}j}^{(q)}\right)} \tag{2.12}
$$

Where $\alpha_{\mathrm{i}j}^{(q)} = a_{\mathrm{i}j}^{(q)}(2\pi)^{\frac{n}{2}}|\Sigma_{\mathrm{i}j}^{(q)}|^{1/2}$ *and the super-index* $(\cdot)^{(q)}$ *denotes the parameter of the* q^{th} *component of the spectral mixture.*

From the kernel expression in eq. (2.12) and the spectral representation in eq. (2.10), the kernel parameters can be interpreted: the cross-spectral delays $\theta_{\mathrm{i}j}^{(q)}$ serves as the time delay between channels; the spectral means $\mu_{\mathrm{i}}^{(q)}$ represent the main frequency of that component, and the cross-spectral means $\mu_{\mathrm{i}j}^{(q)}$ are a weighted sum of the frequencies, each one proportional to the spectral-variances of the other channel, that is, weighting each frequency by the uncertainty of the opposing channel component, making the frequency associated to a low uncertainty component have a higher weight; the cross-spectral phase $\phi_{\mathrm{i}j}^{(q)}$ is simply the difference in phase between channels; the spectral covariance $\Sigma_{\mathrm{i}}^{(q)}$ represents the uncertainty of the distribution in the spectrum, and in the diagonal case is proportional to the inverse of the square root of the lenghtscale of each input dimension, resembling to the automatic relevance determination (ARD) kernel; finally the unnormalized component weights $a_{\mathrm{i}}^{(q)}$ serves as the importance of each component with respect to the total variance of the channel, where the cross magnitude $a_{\mathrm{i}j}^{(q)}$ is proportional to the magnitude of the cross-covariance between channels. The single-channel spectral delay, $\theta_{\mathrm{i}}^{(q)}$ and spectral phase $\phi_{\mathrm{i}}^{(q)}$ do not have a useful interpretation because when $\mathrm{i} = j$ then $\theta_{\mathrm{i}j}^{(q)} = 0$, $\phi_{\mathrm{i}j}^{(q)} = 0$, thus, they only have incidence when comparing different channels.

2.3 Relationship with other models

The proposed MOSM can recover many of the previously proposed models for multi-output Gaussian processes. For instance, if the LMC framework is used with the single output Spectral mixture kernel as covariance functions for the latent process, the denoted SM-LMC [8] kernel is formed, which can be recovered from the MOSM by restricting each spectral mean and spectral covariance to be the same for each channel, and setting delays and phases equal to zero. Moreover, the CONV with a white noise latent function and Gaussian filters

can also be recovered in the MOSM framework if only zero-centred Gaussian are considered in the spectrum, and the delay and phases between channels are set to zero. Likewise, the CSM kernel can be seen as the MOSM kernel with a fixed spectral mean and covariance for each output channel and setting the delays equal to zero. This relationships with previous models are shown in table 2.1, which shows the restrictions to be applied in eq. (2.12) in order to recover previous models.

Model	Restrictions					Degrees of freedom
SM-LMC	$\mu_{\mathrm{i}} = \mu$	$\Sigma_{\mathrm{i}} = \Sigma$	$\theta_{\mathrm{i}} = \theta$	$\phi_{\mathrm{i}} = \phi$	$,\mathrm{i} = 1, \dots M$	$Q(M + 2p)$
CONV	$\mu_{\mathrm{i}} = 0$	-	$\theta_{\mathrm{i}} = \theta$	$\phi_{\mathrm{i}} = \phi$	$,\mathrm{i} = 1, \dots M$	$Q(M + pM)$
CSM	$\mu_{\mathrm{i}} = \mu$	$\Sigma_{\mathrm{i}} = \Sigma$	$\theta_{\mathrm{i}} = \theta$	-	$,\mathrm{i} = 1, \dots M$	$Q(2M + 2p)$
MOSM	-	-	-	-	-	$Q(3M + 2pM)$

Table 2.1: Recovering other MOGP models from the MOSM framework. The parameters mentioned correspond to MOSM formulation in eq. (2.12). In the CSM and SM-LMC kernel, it is assumed the multiplicity of each factor $R_q = 1$, $q = 1, \dots, Q$.

2.4 Alternative Construction: Convolution Process

The MOSM kernel can also be achieved using the convolution process framework [9], where the output functions, $\{f_{\mathrm{i}}\}_{\mathrm{i}=1}^{M}$, are constructed by taking the convolution between independent latent processes or factors denoted, $\{u_q\}_{q=1}^{Q}$ and filters denoted, $\{h_{\mathrm{i}}\}_{\mathrm{i}=1}^{M}$. In the original formulation the filters consisted in real-valued functions, whereas to construct the MOSM kernel, complex-valued filters are necessary.

Considering the latent factors, u_q, to be white noise processes, we aim to build filters h_{i}, such that when convolved with a latent factor, the covariance between the filtered processes results in the MOSM kernel. In order to construct such filters, let for simplicity consider the case of one component, $Q = 1$, where a generalisation is shown at the end of the section, subsequently, recalling that the MOSM kernel can be written as the inverse Fourier transform of the PSD, which by the convolution theorem can be expressed as,

$$\begin{aligned} k_{\mathrm{i}j}(\tau) &= \mathcal{F}^{-1}\left\{S_{\mathrm{i}j}(\omega)\right\}(\tau) \\ &= \mathcal{F}^{-1}\left\{\frac{1}{2}\left(\overline{R_{\mathrm{i}}}(\omega)R_j(\omega) + \overline{R_{\mathrm{i}}}(-\omega)R_j(-\omega)\right)\right\} \\ &= \frac{1}{2}\left[\left(\overline{h_{\mathrm{i}}}(-\tau) \star h_j(\tau)\right)(\tau) + \left(\overline{h_{\mathrm{i}}}(\tau) \star h_j(-\tau)\right)(\tau)\right], \end{aligned} \tag{2.13}$$

where $h_{\mathrm{i}}(\tau) = \mathcal{F}^{-1}\left\{R_{\mathrm{i}}(\omega)\right\}$, and given that R_{i} is a complex-valued Gaussian, the inverse Fourier transform can be found in closed form, where the expression is as follows,

$$\begin{aligned} h_{\mathrm{i}}(x) &= \mathcal{F}^{-1}\left\{R_{\mathrm{i}}(\omega)\right\}(x) \\ &= a_{\mathrm{i}}(2\pi)^{p/2}|\Sigma_{\mathrm{i}}|^{1/2}\exp\left(-(x-\theta_{\mathrm{i}})^{\top}\Sigma_{\mathrm{i}}(x-\theta_{\mathrm{i}})\right)\exp\left(\iota\left((x-\theta_{\mathrm{i}})^{\top}\mu_{\mathrm{i}} - \phi_{\mathrm{i}}\right)\right), \end{aligned} \tag{2.14}$$

the derivation of eq. (2.14) can be found in Appendix.4.6.2. Finally, utilising eq. (1.49) an expression for the output functions can be obtained, taking into account that the covariance between zero-mean complex valued random variables is given by $\text{cov}(f_i(x), f_j(x')) = \mathbb{E}[f_i(x)\overline{f_j}(x')]$, and noting from eq. (2.13) that two convolutions are used to construct the MOSM kernel, then two independent latent processes are used instead of one, which results in the following latent factor construction,

$$f_i(x) = \frac{1}{\sqrt{2}} \left(\overline{h_i}(-x) \star u_1(x) + \overline{h_i}(x) \star u_2(x) \right), \tag{2.15}$$

where u_1, u_2 are independent white noise processes.

Fig. 2.1 shows a example of a sample of the MOSM kernel employing latent factors and convolution process, in top left the latent factor consisting in a white Gaussian noise of unitary variance, top right the filter with real and imaginary part, and bottom the resulting output process which correspond to a channel from a MOGP sample with MOSM kernel. Note that since the filter is complex-valued, convolutions with the latent process results in a complex-valued output as well, so in order to obtain a real-valued output, a transformation must be applied such as taking the real part.

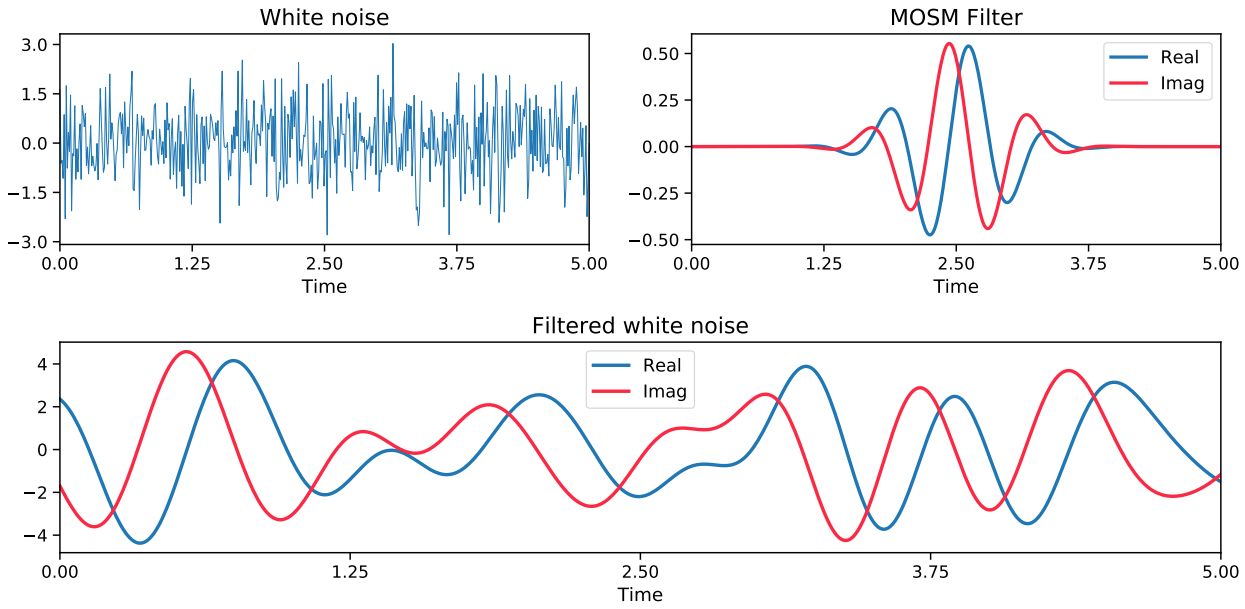

Figure 2.1: Sampling from a mogp with MOSM kernel using the latent factor construction. Top left the white noise sample, top right the MOSM filter, bottom the white noise convolved, which real part correspond to the sample of a MOGP with MOSM kernel.

To corroborate that this formulation recovers the MOSM kernel, the covariance between two functions, f_i, f_j, is obtained, where the white noise processes have covariance $\text{cov}[u_q(x), u_{q'}(x')] =$

$\delta_{qq'}\delta(x-x')$, consequently, the covariance between outputs will be given by,

$$\begin{aligned}
\operatorname{cov}[f_{\mathrm{i}}(x), f_j(x')] =& \frac{1}{2}\mathbb{E}\left\{\left(\overline{h_{\mathrm{i}}}(-x)\star u_1(x) + \overline{h_{\mathrm{i}}}(x)\star u_2(x)\right)\left(\overline{h_j}(-x)\star u_1(x) + \overline{h_j}(x)\star u_2(x)\right)\right\} \\
=& \frac{1}{2}\int \overline{h_{\mathrm{i}}}(-x+z)h_j(-x'+z')\operatorname{cov}[u_1(z)u_1(z')]\,\mathrm{d}z\mathrm{d}z' \\
&+ \int \overline{h_{\mathrm{i}}}(x-z'')h_j(x'-z''')\operatorname{cov}[u_2(z)u_2(z')]\,\mathrm{d}z''\mathrm{d}z''', \quad u_1 \perp\!\!\!\perp u_2 \\
=& \frac{1}{2}\left[\int \overline{h_{\mathrm{i}}}(-x+z)h_j(-x'+z)\mathrm{d}z + \int \overline{h_{\mathrm{i}}}(x-z')h_j(x'-z')\mathrm{d}z'\right] \\
=& \frac{1}{2}\left[\int \overline{h_{\mathrm{i}}}(-\tilde{z})h_j(\tau-\tilde{z})\mathrm{d}\tilde{z} + \int \overline{h_{\mathrm{i}}}(\tilde{z}')h_j\left(-(\tau-\tilde{z}')\right)\mathrm{d}\tilde{z}'\right] \\
=& \frac{1}{2}\left[\left(\overline{h_{\mathrm{i}}}(-\tau)\star h_j(\tau)\right)(\tau) + \left(\overline{h_{\mathrm{i}}}(\tau)\star h_j(-\tau)\right)(\tau)\right]. \qquad (2.16)
\end{aligned}$$

Which is the same expression as eq. (2.13), where $\tau = x - x'$, $\tilde{z} = x - z$, $\tilde{z}' = x - z'$. A generalisation for arbitrary number of components is built summing $Q > 0$ of these structures, that is,

$$f_{\mathrm{i}}(x) = \sum_{q=1}^{Q} \overline{h_{\mathrm{i}}^{(q)}}(-x)\star u_q(x) + \sum_{p=1}^{Q} \overline{h_{\mathrm{i}}^{(p)}}(x)\star u_p(x). \qquad (2.17)$$

Observation This can be interpreted in signals and systems theory as the transfer function, R_{i}, and impulse response h_{i}, of a system, where the white noise u_q serves the role of input of the system. This construction of the MOSM kernel as a convolution process allows for efficient MOGP variational sparse methods [16], where they propose the use of inducing variables as points of the latent factors, this idea is expanded in the following section where inducing variable for MOSM are designed. This construction of MOSM also allows for constructing new multi-output covariance functions, such as non-parametric ones [26].

Chapter 3

Extending the MOSM Kernel

Alongside other MOGP, the MOSM kernel suffers from three important shortcomings, (i) the training of the model does not scale well with the number of data points, making it often unfeasible for large datasets, (ii) given the high number of kernel parameters, the training is prone to fall in local minima, where the optimisation is often sensible to the starting point, and (iii) even with the flexible structure of MOSM kernel, if often fails to regularise when channels are not correlated between them, where uncorrelated channels can hinder the prediction.

In this section said shortcomings of MOSM kernel will be addressed, extending the kernel in various ways. First, we will show how different sparse GP frameworks can be used with MOSM kernel and design inducing variables that better use the structure of the kernel, second, we will construct initialisation methods for the MOSM kernel based on the spectral representation of the kernel, improving the optimisation process, and third, we will propose restricted version of the MOSM kernel which can regularise the number of active components of the kernel, mitigating the negative effects of the uncorrelated channels.

3.1 Alternatives for Sparse Approximations

A key part of the classic and variational sparse formulations for GP are the inducing variables, which serve as a proxy for the training dataset. The usual inducing variables consist in points of the same process where the inducing inputs share the same domain as the original process, although existing work [27] proposes inter-domain inducing variables by taking a linear transformations of the process, and using points of the transformed process as the inducing variables. A direct application is to utilise the Fourier transform of the process to define inducing variables, taking advantage of the spectral content of the process which could be represented in fewer points, where in this case the inducing inputs are in the frequency domain. In practice, as the samples of the process would not be Lebesgue integrable, *ipso-facto* the Fourier transform of the process cannot be obtained, so a windowed version of the process is employed.

Moreover, given the convolution process construction of some kernels, a natural idea would be to use inducing variables consisting in points of the latent processes, where the benefits are twofold, (i) as each latent process is shared across all channels, fewer points are required to represent all outputs, and (ii) as the latent processes are independent, the covariance between inducing variables will contain blocks of zeroes, thus reducing the computational cost.

However, as it was pointed out in previous works [16, 26], considering points of the latent process as inducing variables is unfeasible when the latent process is a white noise process, as it would require an infinite number of points to represent the process. To address this, a linear transformation of the latent processes which consists in filtered versions will be used as inducing variables. Which given that the convolution of the latent process with a filter, causes each point of the filtered white noise to contain information of a neighbourhood of the original process, and possibly all the original process for certain choices of filter. In this section a class of inducing variables utilising the convolution process formulation of the MOSM kernel is proposed, using filtered versions of the latent white noise processes as inducing variables, where this filtered latent processes will be constructed such that, the covariance between inducing variables and covariance between inducing variables and original process can be computed in closed form, this proposed latent inducing variables (LIV) escalate better with the number of channels as each latent component is shared across all channels.

Given a multi-output GP with MOSM kernel a Q components, the proposed inducing variables, denoted λ_q, will be of the form,

$$\lambda_q(z) = G_q(-x) \star u_{q1}(x) + G_q(x) \star u_{q2}(x), \tag{3.1}$$

where z are the inducing inputs, u_{q1} and u_{q2} are independent white noise process for the component q, and G_q is the inducing filter for the latent process. Then, the covariances between inducing variables and the output processes can be obtained in closed form by choosing the inducing filter to be of the form,

$$G_q = a_{\mathrm{i}}(2\pi)^{p/2}|\Sigma_{\mathrm{i}}|^{1/2} \exp\left(-x^\top \Sigma_q x\right), \tag{3.2}$$

which is equivalent as taking the complex conjugate of the MOSM filter defined in eq. (2.14), and setting $\theta_q = 0$, $\phi_q = 0$ and $\mu_q = 0$. To perform inference, the covariance between inducing variables and the covariance between inducing variables and the output must be evaluated, and given the choice of inducing filter as a particular case of the MOSM filter, this quantities can be obtained in closed form, subsequently, the covariance between inducing variables is given by,

$$\begin{aligned}
\operatorname{cov}\left[\lambda_q(x), \lambda_{q'}(x')\right] &= \delta_{qq'}\left(G_q(\tau) \star G_q(-\tau) + G_q(\tau) \star G_q\left(-\tau\right)\right) \\
&= \delta_{qq'}\left(\int G_q(x-z)G_q(x'-z)\mathrm{d}z + \int G_q(x-z)G_q(x'-z)\mathrm{d}z\right) \\
&= a_q^2(2\pi)^{p/2}|\Sigma_q|^{1/2}\exp\left[-\frac{1}{2}\tau^\top\Sigma_q\tau\right],
\end{aligned} \tag{3.3}$$

with $\tau = x - x'$, noting that $G_q(\tau) = G_q(-\tau)$, and the solution of the convolutions follows the same procedure as eq. (2.16). Then, the covariance between the output ith channel and

inducing variable at component q is given by,

$$\begin{aligned}\operatorname{cov}\left[f_{\mathrm{i}}(x), \lambda_q(x')\right] &= \overline{h_{\mathrm{i}}}(-\tau)G_q(\tau) + \overline{h_{\mathrm{i}}}(\tau)G_q(\tau) \\ &= \int \overline{h_{\mathrm{i}}}(-x+z)G_q(x'-z)\mathrm{d}z + \int \overline{h_{\mathrm{i}}}(x-z)G_q(x'-z)\mathrm{d}z \\ &= \alpha_{\mathrm{i}q} \exp\left[-\frac{1}{2}(\tau+\theta_{\mathrm{i}q})^\top \Sigma_{\mathrm{i}q}(\tau+\theta_{\mathrm{i}q})\right] \cos\left[(\tau+\theta_{\mathrm{i}q})^\top \mu_{\mathrm{i}q} + \phi_{\mathrm{i}q}\right], \end{aligned} \tag{3.4}$$

where h_{i} is the MOSM filter defined in eq. (2.14), and similar to the case of the traditional MOSM kernel, the cross parameters in the equation above are given by:

- $\alpha_{\mathrm{i}q} = a_{\mathrm{i}q}(2\pi)^{n/2}|\Sigma_{\mathrm{i}q}|^{1/2}$; $a_{\mathrm{i}q} = a_{\mathrm{i}}^{(q)} a_q \exp\left[-\frac{1}{4}\mu_{\mathrm{i}}^{(q)\top}(\Sigma_{\mathrm{i}}^{(q)} + \Sigma_q)^{-1}\mu_{\mathrm{i}}^{(q)}\right]$
- $\theta_{\mathrm{i}q} = \theta_{\mathrm{i}}^{(q)}$
- $\phi_{\mathrm{i}q} = \phi_{\mathrm{i}}^{(q)}$
- $\mu_{\mathrm{i}q} = \left(\Sigma_{\mathrm{i}}^{(q)} + \Sigma_q\right)^{-1} \Sigma_{\mathrm{i}}^{(q)-1} \mu_{\mathrm{i}}$
- $\Sigma_{\mathrm{i}q} = 2\Sigma_{\mathrm{i}}^{(q)} \left(\Sigma_{\mathrm{i}}^{(q)} + \Sigma_q\right)^{-1} \Sigma_q$.

The proposed inducing variables, LIV, for the MOSM kernel, can be used in sparse methods of the family of reduced rank approximation, which include variational methods, enabling MOSM to be used in large datasets.

A key difference with the classic inducing variables is how cost scales with the number of channels and components, for instance, in the variational sparse approximation, calssic inducing variables, with M number of channels, each one having N points, by considering K inducing points for each channel, the training cost is $\mathcal{O}(NM^2K^2)$ scaling quadratically with the number of channels but begin independent of the number of components of the kernel. Whereas in the proposed LIV, with the same number of channels and observations, but using K inducing points for a kernel with Q components the training cost is $\mathcal{O}(QNMK^2)$ scaling linearly with the number of channels but adding a linear cost in the number of latent components. This trade-off in the proposed inducing variables is useful when working with a higher number of channels and a low number of kernel components, whilst the classic inducing variables can be used in scenarios with low number of channels and higher kernel components.

3.2 Optimisation Considerations

Spectral kernels such as SM, CSM and MOSM although expressive, are difficult to optimise due to the large number of kernel parameters and sensibility present in training, where the optimisation of the NLL is heavily dependent on the starting point. Moreover, given that the number of parameters of said kernels increase with the number of components, adding more components further increases training complexity by increasing the dimension of the optimisation problem. A key factor in the optimisation process is start from a *convenient* initial point, which can greatly assist to avoid falling in local minima, in particular for spectral

kernels there is a way to exploit the frequency domain representation of the kernel to obtain a initial estimation of the hyperparameters utilising the available data prior to optimisation. In the next section two methods will be introduced to facilitate the initialisation of the spectral kernels using an estimate of the power spectral density (PSD) of the channels.

The first consist in estimating the PSD of the available data, then considering said estimate to obtain sound initial parameter values by using the spectral interpretation of the kernel parameters. The second initialisation method is specific to multi-output kernels, where covariance of a given channel is usually a known single output channel, for instance, SM-LMC, CSM and MOSM kernels all reduce to a SM kernel when looking at the covariance of a channel with itself, so a natural way of initialize such models is to fit independent GPs with SM kernels for each channel and then use those parameters to initialize the multi-output kernels. When used in practise, the SM kernels for each channel can be initialized using the first method, yielding a two-stage initialization.

In the next section it we will show that utilising the Bayesian non-parametric spectral estimation (BNSE) [28] to estimate the PSD considering the available observations, can find initial hyperparameters which leads to consistent and better results, compared to previous initialization schemes and other PSD estimations. First the initialisation method based on the PSD estimation will be shown for a single GP with SM kernel, then will be generalized to the MOGP case.

3.2.1 Initializing SM Kernel

Recalling the structure of the SM kernel and the interpretation as a mixture of Gaussians in the spectral domain. In from eq. (1.11), the form of the SM kernel is as follows,

$$k_{SM}(\tau) = \sum_{q=1}^{Q} a_q \exp\left[-\frac{1}{2}\tau^\top \Sigma_q \tau\right] \cos(\mu_q^\top \tau). \tag{3.5}$$

Where in the construction as a multivariate Gaussian in the spectral domain, a_q correspond to the mixture weights, which also can be interpreted as the (unnormalised) magnitude of the spectral component q. The spectral mean of the component, μ_q. And Σ_q the spectral covariance of the component, which in general is diagonal matrix with values $[\sigma_q^{(1)}, \ldots, \sigma_q^{(p)}]$ with p the input dimension. The aim is to find initial values for the aforementioned kernel parameters, using the data available.

Existing initialization schemes choose each parameter individually by some heuristic, in [8], the weights w_q in eq. (1.11) are chosen as constants proportional to the standard deviation of the data, spectral means, μ_q sampled from a uniform distribution from $[0, \eta_{\text{nyquist}}]$ where $\eta_{nyquist}$ is the estimation of the Nyquist frequency given by half of the inverse of the largest interval between input points, and spectral variances, σ_q from a truncated Gaussian with mean proportional to the range of the observation inputs. As each parameter is sampled independently from each other, this method will be denoted Independent Parameter Sampling (IPS) initialisation.

In practice the IPS initialisation yields good but inconsistent results, often requiring a high number of experiments in order to optimise from a adequate starting point and not fall in local minima. As it will be shown in the experiments, for periodic signals it is crucial to obtain a sound initial estimate of the fundamental frequencies, where if misspecified, a common result in the optimisation process is that the kernels reverts to the sum of squared exponential (SE) kernels. That is, the case of SM when the spectral means are zero, $\mu_q = 0$, which greatly affects the forecasting outside the training intervals. As without a periodic component, the SE kernel is known to revert to the prior for predictions far enough (dependant on the lenghtscale) of the training points. In order to address this problem and have a more consistent initialisation, a method is proposed with heavy emphasis in choosing parameters with relation to each other and incorporate the available spectral information from the observations.

The proposed initialization takes as a base the estimated PSD of the process and the number of components Q. The PSD estimate can be obtained from the available spectral estimation literature, such as the periodogram [29], Welch periodogram [30] or Lomb Scargle (LS) [31, 32] among others. To obtain the estimation, the first Q higher peaks in magnitude from the PSD estimate are identified, then, from this peaks: the weights a_q are taken proportional to the magnitude of the peaks, normalised so the sum of the weights is proportional to the variance of the observations (in practice a upscale by a factor of 2 yielded good results),

$$\sum_{q=1}^{Q} a_q \propto \text{Var}(\{y_n\}_{n=1}^{N}), \tag{3.6}$$

where Var() denotes the sample variance; spectral means μ_q as the position of the peaks; and spectral variances σ_q proportional to the width of the peaks (in practice a factor 2 yielded good results). The aforementioned initialization assumes input dimension $p = 1$, but a generalization can be done by estimating individual PSD for each input dimension and assigning accordingly the spectral means μ_q and covariance Σ_q, the weights can be taken as the mean weight across al input dimensions.

Given that in most GP regression problems the observations are not uniformly sampled, the PSD estimation is recommended to be obtained utilising BNSE or LS. For the evaluation of both methods, a grid of frequencies is required, where a uniform grid of frequencies up to the estimated Nyquist frequency is recommended, where the Nyquist frequency can be estimated as half of the inverse of the smallest interval between input points. Initialization using these methods will be refereed as LS initialization and BNSE initialization. In the experiment section, when referring to these initializations word initialization will be omitted for simplicity. A drawback of the proposed spectral initialisation, is that the initialisation will always yield similar results, so in order to include some variability, random Gaussian noise of variance σ^2 will be added to the initial values, this variant of the original initializations will be denoted LS+ϵ and BNSE+ϵ respectively. A summary of the method is shown in the following algorithm,

Algorithm 1: SM Spectral initialisation

Input: Vectors $(\mathbf{x}, \mathbf{y})$ the observations, integer Q the number of components
Result: SM Spectral initialisation
psd $\leftarrow$ calculate_psd_estimate$(\mathbf{x}, \mathbf{y})$;
peaks $\leftarrow$ find_highest_peaks(Q) ;
for $q \leftarrow 1$ **to** Q **do**
    $\mu_q \leftarrow$ position of qth peak ;
    $\Sigma_q \leftarrow$ proportional to the width of qth peak ;
    $a_q \leftarrow$ proportional to the magnitudes of qth peak, following eq. (3.6) ;
end
Output: Kernel parameters, μ_q, Σ_q, a_q, for $q = 1, \ldots, Q$.

3.2.2 Initializing MOSM Kernel

The MOSM kernel defined in eq. (2.12) suffers from a similar problem as the SM kernel regarding the sensibility in the optimization with respect to the initial point, where added to the intrinsic complexity of multi-output GP, results in a even more sensible optimization process. In this context two ways of initialising the hyperparameters prior to the optimization are proposed, following the rationale behind the methods for initialising SM kernel.

The first method for initializing MOSM is an extension of the initialisation of SM kernel based on PSD estimation, where now the PSD will be estimated for each channel, in the case of multiple input dimension, then the PSD estimate is obtained for each channel, for each input dimension. Note that for each output dimension and input dimension, a different estimation of the Nyquist is required.

Let Q be the number of components in the MOSM kernel, with the set of PSD estimates $(\mathrm{PSD_i})_{\mathrm{i}=1}^{M}$, then, for channel i the greater Q peaks are taken for each input dimension, then the magnitudes $a_{\mathrm{i}}^{(q)}$ are assigned as the mean magnitudes for the given channel i then normalized so that the sum of squared weights equals the channel sample variance of the observations of said channel,

$$\sum_{q=1}^{Q} (a_{\mathrm{i}}^{(q)})^2 \propto \mathrm{Var}\left(\left\{y_n^{(\mathrm{i})}\right\}_{n=1}^{N_{\mathrm{i}}}\right), \tag{3.7}$$

and $\left\{y_n^{(\mathrm{i})}\right\}_{n=1}^{N_{\mathrm{i}}}$ correspond to the observations at output i. The spectral means $\mu_{\mathrm{i}}^{(q)}$ are initialised as the position of the peaks; the spectral variances $\Sigma_{\mathrm{i}}^{(q)}$ are set proportional to the width of each peak, in practice the variances are multiplied by 2 so that the uncertainty on a given frequency starts lower and prevent overfit.

The delays $\theta_{\mathrm{i}}^{(q)}$ and phases $\phi_{\mathrm{i}}^{(q)}$ are set to 0 making a initial assumption that there is no input-delay nor phase-delay between channels, leaving to the optimization process to find the non-zero delay and phase if there is.

The second method of initialising consists in fitting individual GP with SM kernel for each channel, given that MOSM kernel in the diagonal is reverted to SM kernel, by fitting

an independent GP with SM for each channel and then utilising those parameters as a initial guess. It is also noted that each individual SM kernel can be initialised with the aforementioned methods in order to further improve the results.

The aforementioned methods can be applied to kernels that are a particular case of MOSM, such as SM-LMC, CONV and CSM by taking into account the restrictions in table. 2.1 to recover such kernels from MOSM.

3.3 Mitigating Negative Transfer of Knowledge

One of the main benefits MOGP is the use of across-channel information to improve predictions by incorporating the information of other channels, this is known as *transfer of knowledge*. However, this transfer of knowledge is only useful if the channels are correlated, and misspecifying this relationship by modelling together unrelated channels could lead to a negative impact in the predictions, where the additional across-channel information leads to a worst prediction than considering the observations of a single channel alone, this phenomenon is called *negative transfer of knowledge*.

To tackle this problem, in [33] they propose an extension of LMC by decomposition each channel in common and specific components, where the common components are shared across all channels, and consequently is where transfer of knowledge occurs, whilst specific components are unique to each output function, modelling individual behaviour of the channels. This decomposition takes the form,

$$f_{\mathrm{i}}(x) = \sum_{q=1}^{Q} a_{\mathrm{i}}^{(q)} u_q(x) + v_{\mathrm{i}}(x), \quad \mathrm{i} = 1, \dots M, \tag{3.8}$$

then, if the processes $\{u_q\}_{q=1}^{Q}$ and $\{v_{\mathrm{i}}\}_{\mathrm{i}=1}^{M}$ are assumed independent, then the covariance kernel is as follows,

$$k_{\mathrm{i}j}(x, x') = \delta_{\mathrm{i}j}\, k_{\mathrm{i}}(x, x') + \sum_{q=1}^{Q} a_{\mathrm{i}}^{(q)} a_{j}^{(q)} k_q(x, x'), \tag{3.9}$$

where k_q is the covariance of the shared process u_q, and k_{i} is the covariance of specific the process v_{i}. This extension of LMC can help mitigate problem of negative transfer of knowledge as each channel has is own "explain away" term to model characteristic of the channel that are not present in other channels, thus not contributing negatively in the prediction.

One limitation only having specific and common components is the lack of control over the influence on a given component, where specific ones only affect one channel and common affect all, with no in between, that is, transfer of knowledge between a subset of channels cannot be achieved in a controlled manner. For instance, say that for component q we only want it shared between channels i and j but not with channel j', by looking at eq. (3.9) one way is by setting $a_{j'}^{(q)} = 0$, but then channel j' would not be able to share information across any other channels using that component q because it was "shut down" for that specific output. A naive solution would be to increase the number of components, adding flexibility

and allowing for each channel to *possibly* have dedicated component by choosing $Q \geq M$ or be capable of modelling the relation of each pair of channels by choosing $Q \geq \binom{M}{2}$, the main drawback of choosing a high number of components is the increased complexity as each component increase the number of kernel parameters to be optimised. In the next section this problem will be addressed considering restricted components, where they can be either pairwise or specific, which in conjunction with the common components allow to increase the flexibility of the kernel while regularising the negative transfer.

Recent work [34] proposes a model to mitigate negative transfer by using non-separable covariances and pairwise modelling, where instead of jointly modelling all the channels, there is a different model for each pair of outputs and the are combined employing the products of GP experts (PoE) [35]. The model also separates between common and specific components, but utilising the non-separable structure of the convolution model [9] instead of a linear combination of latent factors, lastly they use a L1 or L2 regularization on the coefficients of the common processes to further regularise and mitigate negative transfer of knowledge.

Inspired by this, in the following section a restricted version of MOGP kernels will be proposed, being able to adapt components to only influence single, pairs or all components, that when used in conjunction with regularising priors the negative transfer of knowledge can be mitigated.

3.3.1 Restricted-MOSM (R-MOSM)

In order to construct an expressive model whilst mitigating the negative transfer of knowledge, a modification of a non-separable multi-output kernel is proposed, where additional to common and specific components, pairwise components are also incorporate, which only affect a given pair of channels, that way can be controlled (i) individual, (ii) common-to-all channels and (iii) pairwise relationships between channels, and consequently the transfer of information. The pairwise components are constructed such that, for a given pair of channels i, j, the covariance is zero unless they are the pair corresponding to said component, that is,

$$k_{\mathrm{i}j}^{\mathrm{i}'j'}(x, x') = \begin{cases} k_{\mathrm{i}j}(x, x') & \text{if } (\mathrm{i}, j) = (\mathrm{i}', j') \text{ or } (\mathrm{i}, j) = (j', \mathrm{i}') \\ 0 & \text{otherwise} \end{cases}. \tag{3.10}$$

Denoting Q_s, Q_c, Q_p the number of specific, common and pairwise components respectively, the covariance using a mix of the three components will be given by,

$$k_{\mathrm{i}j}(x, x') = \delta_{\mathrm{i}j} \sum_{q_s=1}^{Q_s} a_{\mathrm{i}}^{(q_s)} k_{\mathrm{i}}^{(q_s)}(x, x') + \sum_{q_c=1}^{Q_c} a_{\mathrm{i}}^{(q_c)} a_j^{(q_c)} k_{\mathrm{i}j}^{(q_c)}(x, x') + \sum_{q_p=1}^{Q_p} a_{\mathrm{i}}^{(q_p)} a_j^{(q_p)} k_{\mathrm{i}j}^{(q_p)}(x, x'). \tag{3.11}$$

Furthermore, a regularising prior such a Gaussian or Laplace can be imposed on the common and pairwise component coefficients $a_{\mathrm{i}}^{(q)}$ to further reduce the negative transfer and prevent overfit due to the increased number of components. This type of restricted component can be used in any kind of non-separable multi-output kernel, where for M channels, this restricted approach can be seen as choosing $Q = M \cdot Q_s + \binom{M}{2} Q_p + Q_c$ components, while restricting some weights $a_{\mathrm{i}}^{(q)} = 0$ accordingly. In particular, by choosing all the components eq. (3.11) to

be the MOSM kernel, will be denoted Restricted-MultiOutput Spectral Mixture (R-MOSM) kernel.

Chapter 4

Validation

4.1 Experiment setting

To validate MOSM and the proposed improvements, five experiments are shown. The first one compares the different initialisation methods proposed for spectral kernels, using a single output GP regression considering CO2 concentration data. The second experiment, regarding the negative transfer of knowledge, consists in a synthetic dataset where the negative transfer can be controlled, comparing the restricted MOSM against regular MOSM. The third experiment shows the sparse capabilities of MOGP by comparing the proposed latent inducing variables for MOSM against regular inducing variables in a robot inverse-dynamic dataset with 44000 points. The last two experiments consist in finance time series applications using MOGP, where the MOSM kernel is compared against previous models.

As error metrics for the first three experiments, the root mean squared error (RMSE), standardised mean squared error (SMSE) and negative log predictive distribution (NLPD) were used as performance measurements. When the outputs are in the same scale the RMSE and NLPD were used, whilst the SMSE was obtained when the outputs are in different scales, which is common in the multi-output case as different channels can have unique scales. Moreover, denoting $\{y_n\}_{n=1}^N$ the training points, σ the variance of the test points and $\{y_n^*\}_{n=1}^N$ the predictive mean of the model, the RMSE and SMSE are defined as,

$$\text{RMSE} = \left(\frac{1}{N} \sum_{n=1}^{N} (y_n - y_n^*)^2 \right)^{1/2}, \tag{4.1}$$

$$\text{SMSE} = \frac{1}{N} \sum_{n=1}^{N} \frac{(y_n - y_n^*)^2}{\sigma^2}. \tag{4.2}$$

These two measurements are representative of point prediction error, a key difference between the two is that the RMSE is in the same scale as the output, whereas the SMSE is normalised by the variance of the test set so it can be compared between outputs of different scale. The SMSE also has the property that, in the case of predicting with the naive model consisting of the mean value, the SMSE would be equal to 1, making it easy to interpret, as values higher

than 1 means the model is performing worse than the taking the mean of the test set, which in the case of GP when the data is usually normalised to have zero mean, indicates that the model was successfully trained, or when values of SMSE near 1, the posterior reverted mainly to the prior.

Finally, the NLPD is a measure of the distribution prediction error, defined as the negative log probability of the test data, given the learnt model,

$$\text{NLPD} = -\frac{1}{N}\sum_{n=1}^{N}\log\left(p_n(y_n)\right), \tag{4.3}$$

where $\{p_{\text{i}}(y_{\text{i}})\}_{n=1}^{N}$ is the learnt predictive distributions. All metrics shows a better performance the lower the value.

All models, including the proposed MOSM are available in the mogptk toolkit which will be briefly described in the following section. All the experiments are available [1].

4.2 MOGPTK: Multi-output GP Toolkit

In order to compare the proposed MOSM against existing MOGP kernels as well as to have framework to incorporate the proposed initialisations, optimisation options and improvements on the MOSM kernel, we built a Multi-Output Gaussian Process Toolkit (MOGPTK) [5] as Python package for training multi-channel datasets using Gaussian processes. It extends GPflow [36], a general Gaussian process Python library, that in turn is built upon TensorFlow [37] allowing GPU accelerated training. This toolbox was developed alongside Taco de Wolff, while working on this thesis.

MOGPTK implements four popular MOGP models: the Linear Model of Corregionalization (LMC), the Cross-Spectral Mixture (CSM), the Convolutional Model (CONV), and the Multi-Output Spectral Mixture (MOSM). The toolkit facilitates implementing the entire pipeline of GP modelling, including data loading, parameter initialization, model learning, parameter interpretation, up to data imputation and extrapolation.

The toolkit also includes the three main contributions of this paper, (i) the initialisation methods for spectral multi-output kernels in section 3.2, to increase the likelihood of convergence and speed up the training process, (ii) the restricted-MOSM which helps to mitigate the negative transfer of knowledge, with individual and pairwise components, and (iii) the latent inducing variables for sparse approximations which escalate better with the number of channels. MOGPTK also allow to manually fix parameters before training, enabling flexible training, where parameters can be optimised independently in stages.

Besides training and prediction, MOGPTK also provides interpretation of hyperparameter values via visualization techniques. MOGPTK shows the correlations between channels for different kernels, in the particular case of spectral kernels (e.g., SM, MOSM, CSM, SM-LMC), this reveals the cross-spectral coupling between channels. Finally, MOGPTK features

[1] `https://github.com/Ale-Cuevas/msc_thesis`

general-purpose classes to perform common data-analysis operations effortlessly. Data can be easily loaded from various sources (e.g., CSV files, Pandas DataFrames, or generated using Python functions) and can also be pre-processed utilising included transformations such as detrending or logarithm among others. Additionally, MOGPTK allows for removing data ranges to simulate missing data or sensor failure and the data can be easily plotted in time or spectral domain.

MOGPTK is a complete package for MOGP training and is freely available for both open-source and commercial applications. The source code, tutorials and examples in the form of Jupyter notebooks, together with the API documentation, can be found at `http://github.com/GAMES-UChile/mogptk`.

4.3 Comparing initialisations

To compare the proposed initialisations, an experiment was performed in a single channel GP regression setting using the SM kernel, considering four different optimisation methods and six initializations. The initialization methods being: Random, IPS, LS, LS+ϵ, BNSE, BNSE+ϵ, where the noise added in BNSE and LS case were Gaussian with variance equal to 1/30 of the value of the parameter. The random initialization samples each parameter from a uniform in $(0, 1)$, the LS based initialisations was done evaluating the periodogram in a similar uniform grid of 50000 points from 0 to the estimated Nyquist frequency, lastly the BNSE based initialisations were evaluated using a uniform grid of frequencies of 10000 points, from 0 to the estimated Nyquist frequency. The Nyquist frequency was estimated as half of the inverse of the minimum distance between inputs, that is $\xi_{\text{Nyquist}} = 0.5 \cdot \frac{1}{\text{d}_{\text{min}}}$ where d_{min} is the minimum distance between inputs.

The optimisation methods used were: conjugated gradient (CG), L-BFGS-B and Adam. We compared all combination of initialisation and optimizers considering the monthly CO2 concentration at Mauna Loa observatory ([38]) consisting of 521 monthly observations, using the first 300 observation as training and the remaining as test. The data was normalised so it has unit zero mean and unit variance, then, for each pair of initialisation-optimizer, a GP regression with SM kernel, utilising $Q = 10$ was trained. For each model, training time and number of function evaluations was compared as well as RMSE and NLPD in test set. For L-BFGS-B and CG a maximum of 2000 iterations and tolerance of 10^{-6} was used, for Adam 2000 iterations were used with learning rate set to 0.01 and remaining parameters to default value.

The results averaged over 10 trials are shown in table. 4.1, whereas the training times and number of NLL evaluations are shown in table. 4.2. The proposed spectral initialisation methods are able to find good starting points for the optimisation, where the BNSE initialisation performed well across all optimisation methods, obtaining the lower NLPD in all optimisation methods, and obtaining the lower RMSE when employing L-BFGS-B and CG, where for Adam the lower RMSE is obtained using the LS initialisation.

	L-BFGS-B		CG		Adam	
	RMSE	NLPD	RMSE	NLPD	RMSE	NLPD
Random	4.063 ± 0.916	3.217 ± 0.683	4.738 ± 0.015	4.330 ± 0.373	4.151 ± 0.133	9.447 ± 1.424
IPS	10.563 ± 8.096	4.011 ± 1.672	4.986 ± 5.311	16.305 ± 38.282	4.308 ± 5.330	6.644 ± 10.81
LS	3.420 ± 0.000	3.297 ± 0.000	3.808 ± 0.000	6.578 ± 0.000	$\mathbf{2.952 \pm 0.000}$	6.126 ± 0.000
LS+ϵ	4.042 ± 1.224	3.438 ± 1.366	2.912 ± 1.338	15.937 ± 30.688	3.340 ± 1.288	29.295 ± 32.209
BNSE	$\mathbf{2.269 \pm 0.000}$	$\mathbf{1.923 \pm 0.000}$	$\mathbf{0.751 \pm 0.000}$	$\mathbf{1.441 \pm 0.000}$	3.659 ± 0.000	$\mathbf{5.734 \pm 0.000}$
BNSE+ϵ	4.222 ± 2.347	2.770 ± 0.755	3.613 ± 1.374	6.206 ± 6.303	3.091 ± 0.854	7.305 ± 7.954

Table 4.1: RMSE and NLPD for different initialisations and optimisers in GP regression using SM for Mauna Loa dataset, results averaged over 10 trials.

Looking at the training times and number of NLL evaluations, the main factor is the optimisation method rather than the initialisation, where it can be seen that the higher number of NLL evaluations of CG correlates with a better performance in the estimation.

	L-BFGS-B		CG		Adam	
	Time [s]	NLL evals	Time [s]	NLL evals	Time [s]	NLL evals
Random	1.8	84	74.7	3488	46.9	2000
IPS	5.7	268	85.1	3965	46.9	2000
LS	1.8	83	82.6	3852	47	2000
LS+ϵ	2.4	114	86.9	4054	46.9	2000
BNSE	2.3	105	92	4286	47	2000
BNSE+ϵ	3.3	153	87.5	4076	47	2000

Table 4.2: Time and number of objective function evaluation for different initialisations and optimisers in GP regression using SM for Mauna Loa dataset, results averaged over 10 trials.

The proposed spectral initialisations were able to estimate sound initial parameters for the SM kernel, decreasing the likelihood of falling in local minima. By using the PSD estimate and the spectral representation of the kernel, the method computes initial kernel parameters which incorporate the available information into the kernel parameters. In particular, initialising the kernel parameters using BNSE for the PSD estimate yielded the higher performance results, BNSE was able find the position of the peaks which corresponded to the frequencies with higher energy, and not overfitting by assigning mass to nearby frequencies, resulting in peaks with a non-zero width. Where the width of the peaks the mixture of Gaussians setting represent the uncertainty of a component, this allows the optimisation process the perform a broader search, while still starting from an advantage starting point.

A drawback the proposed spectral methods is that the initialisation is deterministic, thus showing no variance in the results, given that the optimisation method is deterministic, which can be detrimental if the initialisation does not yield a good result. This can happen if the sinusoidal components of the data are not predominant, whereas the noisy variant of the LS and BNSE show decreased performance, but also show variability in the results, allowing to maintain the spectral nature of the initialisation while yielding different initial points for the optimisation. Another drawback of the proposed initialisations comes when utilising a high

number of input dimensions, where the proposed methods yield insufficient results, mainly due to the nature of the Fourier transform of a function with a multidimensional domain, where obtaining PSD independently by each input dimension is not enough to represent the spectrum of the function. This can be understood when interpreting the spectrum of a function as a distribution, where estimating the PSD independently by each axis is the same as estimating the marginals of a multivariate distribution, and by the marginals alone the joint distribution cannot be reconstructed, which is the same for the Fourier transform of a function with domain of dimension $p > 1$.

4.4 Synthetic example of negative transfer

To assert that the proposed Restricted-MOSM (R-MOSM) is able to mitigate negative transfer of knowledge, a synthetic example is constructed where the negative transfer can be controlled, consisting in a asymmetric multi-output regression. The asymmetry relates in that there exist some hierarchy in the relevance of the outputs, that is, some channels are more important in terms of obtaining good prediction than others. In this case the main interest will be obtaining predictions on a single channel with relatively low number observations, while having other channels with more observations to help (or not) in the prediction. The channel of interest will be denoted primary channel or output whilst the remaining will be denoted secondary channels.

In order to control the negative transfer of knowledge, given a fixed number of secondary channels, they will be constructed in such way that they will or will not be related with the primary channel, thus controlling the negative transfer with the number of non-related channels, if the number of non-related outputs is high, then it is more likely for the model prediction on the primary channel to be hindered by the cross-channel observations.

The experiment consist in one primary channel, f_p, sampled from a GP with zero mean and covariance given by a primary kernel SE, K_{SE}^{prim}, with lengthscale, $\ell_p = 0.5$ and unitary variance, then the sample is corrupted with Gaussian noise, ϵ_p, of scale equal to $\sigma_{np} = 0.3$, from this corrupted version, $y_p = f_p + \epsilon_p$, the observations will be generated. Then 4 secondary channels will be constructed, which can be related or unrelated to the primary channel, first, related channels, denoted y_{ir}, are sampled from a GP with mean equal to the uncorrupted primary channel and covariance given by a secondary kernel SE, K_{SE}^{sec} with lengthscale, $\ell_s = 0.1$, and unitary variance, additionally the sample is multiplied by a random coefficient a sampled uniformly from $[0, 1]$, lastly the channel is corrupted with Gaussian noise of scale equal to, $\sigma_{nr} = 0.05$,

$$\begin{aligned} &f_{\text{ir}} \sim \mathcal{GP}\left(f_p, K_{SE}^{\text{sec}}\right); \quad a \sim U(0,1); \quad \epsilon_{\text{ir}} \sim \mathcal{N}(0, \sigma_{nr}^2) \\ &y_{\text{ir}} = a\, f_{\text{ir}} + \epsilon_{\text{ir}}. \end{aligned} \tag{4.4}$$

Then, for constructing unrelated channels, denoted y_{iu}, first a base unrelated function is obtained, f_u, sampled from a GP with zero mean and covariance given by a unrelated kernel SE, K_{SE}^{unrel}, with lengthscale $\ell_u = 1$, and unitary variance, subsequently the unrelated channels are constructed in the same way as the related, but this time the mean of the GP sampled

is f_u, that is,

$$f_{iu} \sim \mathcal{GP}\left(f_s, K_{SE}^{\text{sec}}\right); \quad a \sim U(0,1); \quad \epsilon_{iu} \sim \mathcal{N}(0, \sigma_{nu}^2)$$
$$y_{iu} = a\, f_{iu} + \epsilon_{iu}, \tag{4.5}$$

where noise scale shares the same value as the related channels, $\sigma_{nu} = 0.05$. Finally, for a given multi-output process consisting in the primary channel and 4 secondary channels which can be related or not the primary, the observations are obtained by sampling 150 equispaced points from $[0, 10]$. Then, first 80% of the primary task data was randomly removed, this removed points were same across all trials, after that, the number of related and unrelated channels was varied, starting with 0 related and 4 unrelated channels, and increasing the related and decreasing the unrelated until there are 4 related and 0 unrelated.

This was repeated 5 times, in each one fitting 3 models, (i) regular MOSM with $Q = 1$, (ii) a R-MOSM with $Q_s = 1$ and $Q_u = [0, 1, 1, 1, 1, 1]$, $Q_p = 0$ that is, there is a regular MOSM component shared across all channels and each secondary channel has an individual component, but no the primary channel, and (iii) a R-MOSM with $Q_s = 1$, $Q_u = [0, 0, 0, 0, 0, 0]$, $Q_p = 1$, that is, a shared component across all channels and a component for each pair of channels only influencing said pair, this model will be denoted R-MOSM-P. As a baseline, a single output GP was fitted only using the primary channel observations, with a SM kernel, this will be used to assert if there exist negative transfer in the learning, where if the model performs worse than considering only the primary channel observations, negative transfer is present.

For the primary channel, in Fig. 4.1 and Fig. 4.2 is shown the RMSE and NLPD respectively, in function of the number of related secondary channels, showing the mean and standard deviation of the 5 trials.

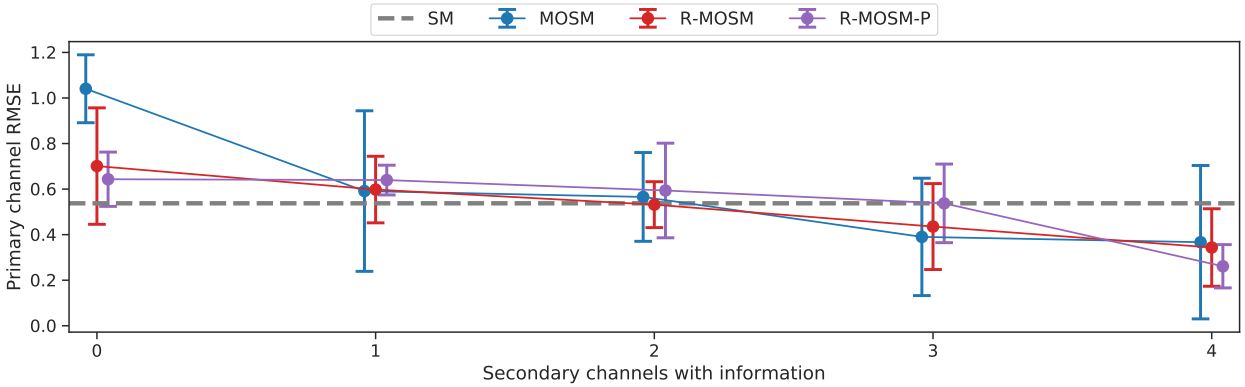

Figure 4.1: RMSE in the primary channel in function of the number of secondary channels that share information with the primary.

By looking at the RMSE plot in Fig. 4.1, the negative transfer can be seen clearly when all 4 channels are uncorrelated, where the classic MOSM becomes more hindered in the prediction than the restricted counterparts, with the increasing number of correlated channels the negative transfer decreases, being mitigated around 2 correlated channels onwards. It

can be seen that at 4 correlated channels, classic MOSM presents a higher error compared to R-MOSM and R-MOSM-P, where the confidence interval falls above the baseline.

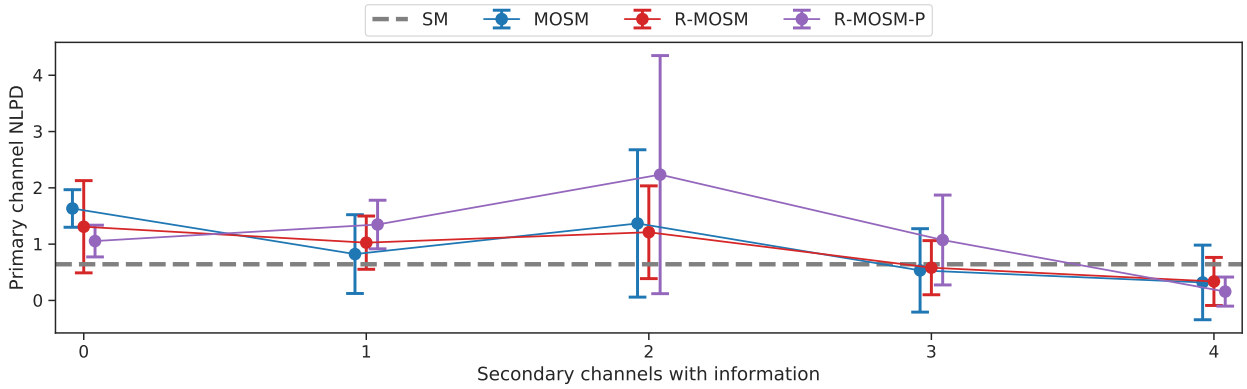

Figure 4.2: NLPD in the primary channel in function of the number of secondary channels that share information with the primary.

From the NLPD plot in Fig. 4.2, it can be seen that all models present a degree of negative transfer when at least one channel is uncorrelated, one possible explanation is, given that the RMSE falls under the baseline at 2 correlated channels, is that the models show high uncertainty in the prediction, thus making the predictive variance take high values. The tables with the values of the plots can be seen in appendix 4.6.2.

Fig. 4.3 shows the prediction on the primary channel for the baseline model, single output GP with SM kernel, it can be seen that the prediction around $x = 5$ is not able to reconstruct the curve of the target function, where the target function lies outside the confidence interval of the prediction, the observations are not enough for the GP to be able fit correctly the target function. In contrast, the prediction by R-MOSM shown in Fig. 4.4, is able to reconstruct the target function following it closely, where cross-channel information compensates the lack of data of the primary channel.

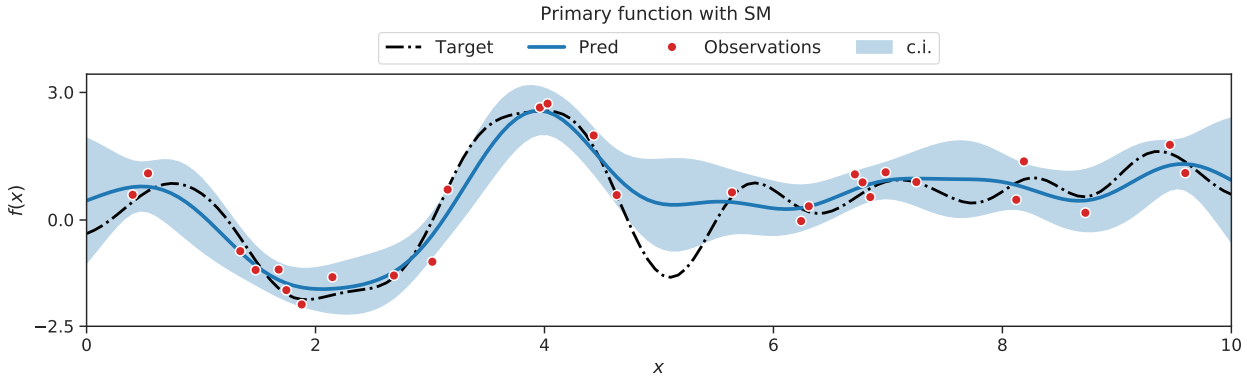

Figure 4.3: Regression on the primary channel using single output GP with SM kernel.

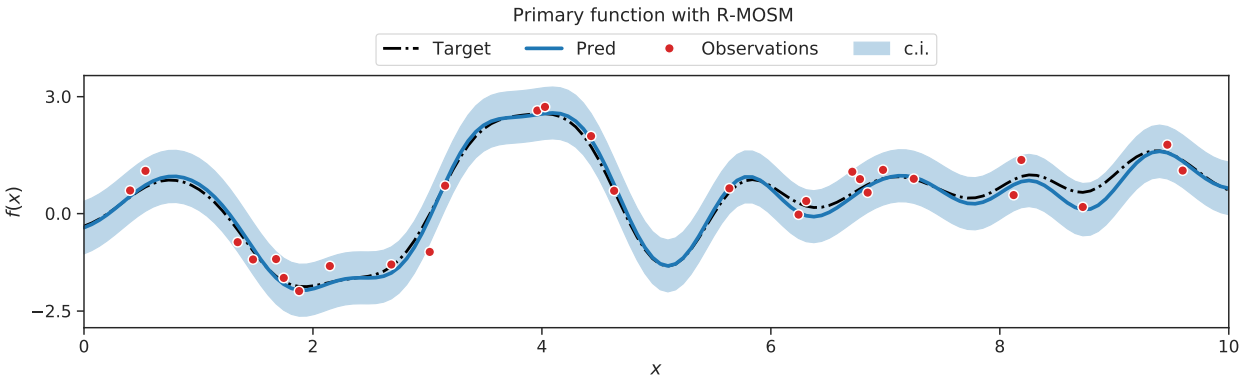

Figure 4.4: Regression on the primary channel using R-MOSM.

For R-MOSM, pairwise components allows to learn relationships which are only shared across pairs of channels, while individual components are able to model specific behaviour of a single channel, therefore preventing the transfer of information to other uncorrelated channels. In contrast, regular MOSM forces relationships among channels, as the optimisation process is unconstrained. A drawback of the proposed R-MOSM is the increased complexity in training, given the sparse structure of the kernel, and the need to chose the parameters of regularising priors.

The proposed R-MOSM was able to mitigate the negative transfer, reducing the error when uncorrelated channels were present. However, this is clear only when there is 0 to 1 correlated channels, where by increasing the number of correlated channels shortens the gap between the errors of MOSM and R-MOSM. This is due to the non-separable nature of MOSM kernel, where given a high enough number of components the MOSM kernel is able to naturally mitigate some of the negative transfer.

Unfortunately, both the R-MOSM and MOSM were not able to fully mitigate the negative transfer, still yielding higher error than the baseline when no correlated channels were present.

4.5 Robot Inverse Dynamic Problem

The robot inverse dynamic problem is an important area in robotics, where the objective is usually to obtain the forces or torques of a robot motors based of the kinematics, such as the positions, velocities and accelerations. A commonly dataset used in multi-output regression is the SARCOS dextrous arm dataset [39], which has been tackled previously using sparse multi-output GP in [40, 41], the dataset relates to the dynamic model of a seven degree-of-freedom robot arm, where the problem consist in estimating the 7 torques utilising the 21 inputs composed of the 7 positions, velocities and accelerations of the joints, consisting in a total of 48933 observations.

We tackle the problem of learning the inverse dynamic problem employing the MOSM

kernel, where in order to cope with the high number of observations, we employed the variational sparse multi-output approximation with stochastic variational inference (SVI), which allows to work in mini-batches, in conjunction with the proposed inducing variables, this model will be denoted SVI-MOSM-LIV.

For comparison, we also fitted MOGP using SVI with MOSM kernel and inducing variables in the input domain; and independent GP with SM kernel and SVI sparse approximation, which will be denoted SVI-MOSM and SVI-IGP-SM respectively. From the total of data available, 44484 points where selected as the train test and the remaining 4449 for testing, the joint learning of two couples is considered, the 2nd and 3rd torques which are negatively correlated, and the 4th and 7th torques which are positively correlated [39].

All models were trained with $Q = 1$, and as was mentioned in the experiment regarding the spectral initializations, they prove to be unsatisfactory for a high number of input dimensions, so all kernels parameters are initialized by sampling from a uniform in $[0, 1]$ with the exception of the phase and delay which are initialized to zero. For all the models, 800 inducing points were used, for SVI-MOSM and SVI-IGP-SM the inducing locations were initialized by choosing randomly from the training set. For SVI-MOSM-LIV the inducing locations are also initialized by randomly choosing from the training inputs, where the difference with SVI-MOSM is that, as the inducing points are in the latent process, it does not need to consider the channel, making that each inducing point affects all the channels equally for that component.

A desired property when working with mini-batches, is that the mean ELBO of the mini-batches approximates the ELBO of the complete dataset, however this cannot be tested as the size of the dataset makes prohibitive the evaluation of the ELBO using all the observations, to assert this, instead 7000 random points were selected, from this, 600 mini-batches of 1200 points each were obtained, in each one evaluating the ELBO, the histogram of the evaluations was obtained is shown in Fig. 4.5 alongside the mean of the evaluations.

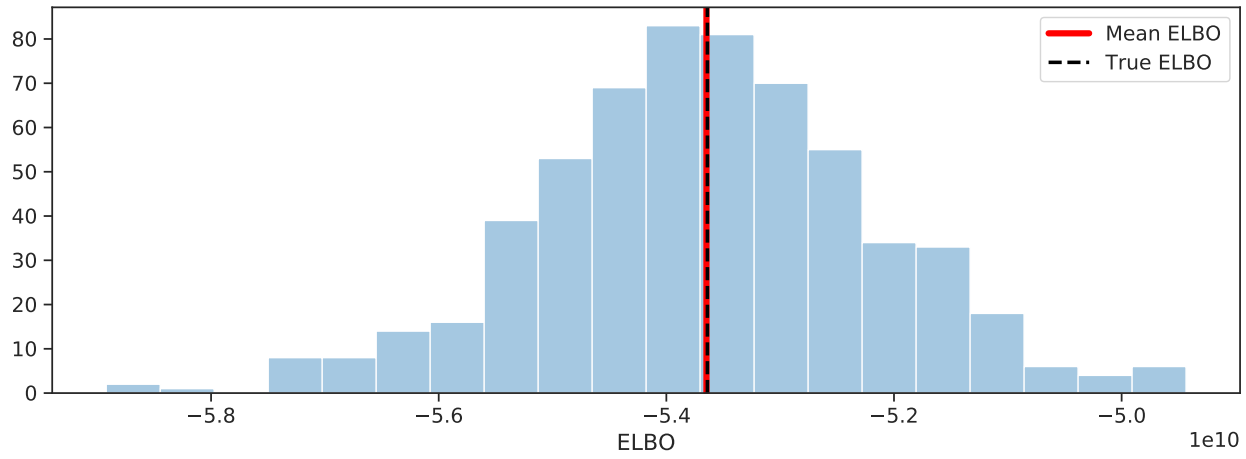

Figure 4.5: Histogram of ELBO of mini-batches for the 2nd and 3rd joint using a subset of 7000 points, in red line the mean of the evaluations, in black dashed line the ELBO of the whole subset.

Each experiment was run considering 800 inducing points, a batch size of 1200 and running 2000 iterations with Adam optimiser , where due to the large number of parameters, the optimisation was done in fixing some parameters alternating between training the kernel and the inducing inputs position and parameters if it is the case.

The first 500 iterations the kernel was trained leaving the inducing location fixed, the next 500 iterations the inducing variables, then the following 500 the kernel again and the last 500 iterations the inducing inputs were trained. The learning rate for the first cycle was set to 0.05 and the following to 0.1. Training times were in the same order of magnitude, with SVI-MOSM-LIV taking 15 minutes for training, SVI-MOSM 21 minutes and SVI-IGP-SM 30 minutes. For each model 5 trials were run.

For SVI-MOSM-LIV considering the learning of 4th and 7th joint, in Fig. 4.6 is shown the ELBO of the mini-batches in training, where in blue is the training of kernel parameters and in red the training of the inducing location, it can be seen that the first cycle of kernel training does not increase the ELBO significantly, whilst the remaining cycles show a steady increase in the ELBO. It is also worth noting the oscillating behaviour in the whole training process, suggesting the use of a lower learning rate.

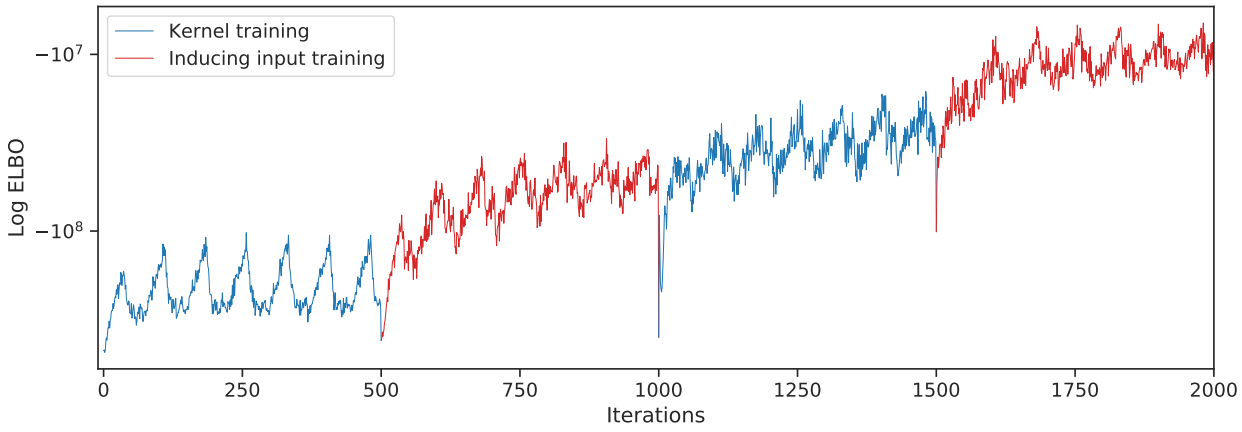

Figure 4.6: Evolution of mini-batches ELBO in training, in blue the kernel training, in red the inducing variables location training.

The results averaged over the 5 trials are shown in table. 4.3 for torques 2 and 3, for the 2nd torque, all models stays in the same order of magnitude, where SVI-MOSM presents the lower RMSE and SMSE, indicating a better prediction of the posterior mean, whilst in torque 3 the independent GP with SM kernel show the lower RMSE and SMSE. But when looking at the NLPD is clear that the SVI-MOSM and SVI-IGP-SM tend to overfit in the learning process, where the proposed SVI-MOSM-LIV is able to regularize the posterior showings the lower NLPD in both torques, where in the other cases the predictive variance is too low.

A possible explanation is that, given that the proposed LIV inducing features are shared across all channels, in the optimisation process the inducing locations would be positioned so it benefits all channels, thus reducing overfitting, as with independent inducing locations

for each channels the positions could be adjusted to only benefit the corresponding channel.

Table. 4.4 shows the results for for torques 4 and 7, were similar to the other pair, for the torque 4, the SVI-MOSM shows the lower RMSE and SMSE, where for torque 7 the SVI-IGP-SM shows the lower RMSE and SMSE, but when evaluating the NLPD is clear that both SVI-IGP-SM and SVI-MOSM tend to overfit, where the SVI-MOSM-LIV shows the lower NLPD.

	2nd joint			3rd joint		
	RMSE	SMSE	NLPD	RMSE	SMSE	NLPD
SVI-IGP-SM	2.852 ± 0.115	0.037 ± 0.003	306.986 ± 147.337	$\mathbf{1.44 \pm 0.079}$	$\mathbf{0.021 \pm 0.002}$	46.253 ± 54.181
SVI-MOSM	$\mathbf{2.572 \pm 0.251}$	$\mathbf{0.030 \pm 0.006}$	63.045 ± 80.303	1.96 ± 0.926	0.047 ± 0.048	424.204 ± 662.538
SVI-MOSM-LIV	3.032 ± 0.209	0.042 ± 0.006	$\mathbf{11.987 \pm 4.598}$	1.855 ± 0.149	0.035 ± 0.006	$\mathbf{12.133 \pm 5.653}$

Table 4.3: RMSE, SMSE and NLPD for 2nd and 3rd joint in robot inverse dynamics problem, results averaged over 5 trials.

	4th joint			7th joint		
	RMSE	SMSE	NLPD	RMSE	SMSE	NLPD
SVI-IGP-SM	3.647 ± 5.043	0.205 ± 0.396	335.33 ± 630.374	$\mathbf{0.403 \pm 0.064}$	$\mathbf{0.025 \pm 0.008}$	14.631 ± 7.757
SVI-MOSM	$\mathbf{1.270 \pm 0.107}$	$\mathbf{0.009 \pm 0.001}$	34.774 ± 63.767	0.827 ± 0.777	0.191 ± 0.324	275.196 ± 518.288
SVI-MOSM-LIV	1.464 ± 0.202	0.012 ± 0.003	$\mathbf{4.981 \pm 4.306}$	0.488 ± 0.048	0.036 ± 0.007	$\mathbf{4.436 \pm 2.316}$

Table 4.4: RMSE, SMSE and NLPD for 4th and 7th joint in robot inverse dynamics problem, results averaged over 5 trials.

The proposed LIV can be integrated with existing sparse approximations, allowing to train using a large number of observations. When used in conjunction with stochastic variational sparse approximation, a MOGP can be trained in 15 minutes using 44000 points, while classic inducing variables takes 20. Classic inducing variables must be defined per output, whereas the proposed LIV are shared across all channels, this allows the use of fewer inducing variables to represent the whole process, and better scalability with the number of channels, although at a higher cost with respect to the number of components. This trade-off suggest which inducing variable to use depending on the application, employing the proposed LIV when working with a high number of channels but few kernel components, and classic inducing variables when the number of channels is low and the number of components high.

A key assumption of the proposed LIV is that all channels are correlated, given that the inducing variables correspond to points of the latent processes. If the channels are not correlated and the LIV are utilised, when optimising the inducing locations the result may no be sufficient to represent the whole process, where some inducing location could only encapsulate useful information for certain channels. In this case, a higher number of inducing variables must be used to incorporate the different structures, another option is use a mixed set of inducing variables, mixing the LIV with classic inducing variables.

4.6 Finance time series applications

The following two sections show applications of MOGP with MOSM kernel in finance datasets, the first estimating missing values in a four channel model with observations of the price of gold, oil, the compound index NASDAQ and a USD index. The second consist in the exchange rate of ten different countries with respect to USD, where the imputation of missing values is considered. Both scenarios form part of a work presented in [24].

For the finance time series experiments, the error metrics employed were the mean-normalised mean absolute error (nMAPE) and mean-normalised root mean squared error (nRMSE),where denoting N the number training points, $\{y_n\}_{n=1}^N$ the observations and $\{y_n^*\}_{n=1}^N$ the predictive mean of the model at training locations, the error metrics are defined as follows,

$$\text{nMAPE} = \frac{1}{N}\sum_{n=1}^{N}\frac{|y_n - y_n^*|}{\bar{y}} \cdot 100 \tag{4.6}$$

$$\text{nRMSE} = \left(\frac{1}{N}\sum_{n=1}^{N}\frac{(y_n - y_n^*)^2}{\bar{y}}\right)^{1/2}, \tag{4.7}$$

$$\tag{4.8}$$

with, $\bar{y} = 1/N\sum_{n=1}^{N} y_n$, the mean of the observations.

4.6.1 Gold, Oil, NASDAQ, and USD index

We considered a dataset comprising series of gold and oil prices, the NASDAQ and the USD index (henceforth referred to as GONU) [42, 43, 44], between January 2017 and December 2018 with a weekly granularity. We detrended and log-transformed the data signals and removed regions in each channel to mimic missing data. For oil we removed observations between 2018-10-05 and 2018-12-31 as well as removing 30% of all observations randomly. For gold we removed observations between 2018-07-01 and 2018-10-01. Finally, for the gold, NASDAQ and USD channels we removed 60% randomly. Overall, our experiment consisted of 385 training points and 446 test points resulting in roughly five minutes of training time for the MOSM. We also set a Gaussian prior on the covariance magnitudes with the standard deviation of the hyperparameter set to the maximum value of each channel.

Fig. 4.7 shows a fit of the MOSM kernel. The MOSM model is able to encapsulate the structure of the channels with almost all data within the confidence interval of 95%, even for parts that have missing data but with a deviating imputation for NASDAQ. The related cross-correlation matrix is plotted in Fig. 4.8. Notice that the empirical cross-correlation matrix is showing correlation between gold, oil, and NASDAQ, with especially a strong dependency between oil and NASDAQ thus confirming our hypothesis. The hedging quality of gold can also be seen (albeit faintly) with the negative cross-correlation between gold and the USD index.

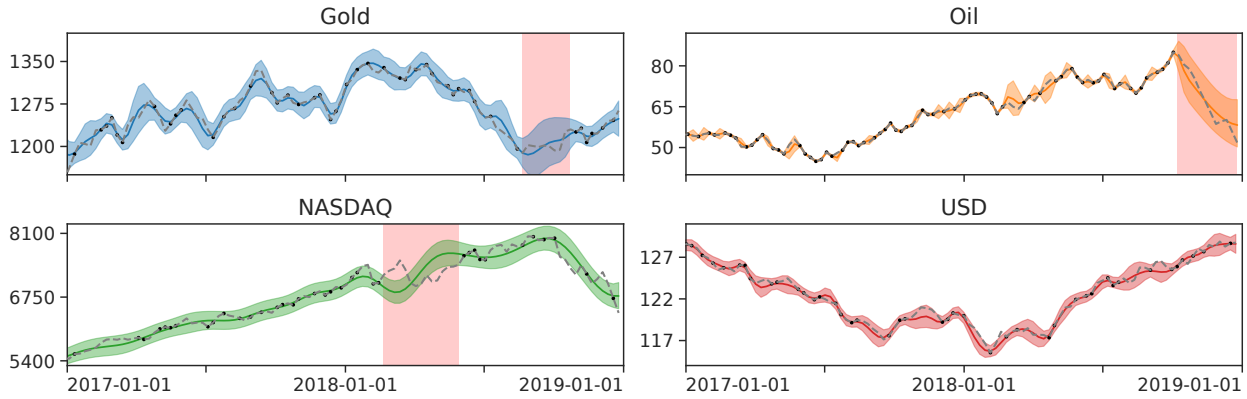

Figure 4.7: GONU data set with the trained MOSM kernel. Training points are shown in black, dashed lines are the ground truth and the colour coded lines are the posterior means. The coloured bands show the 95% confidence intervals. The red shaded areas mark the data imputation ranges.

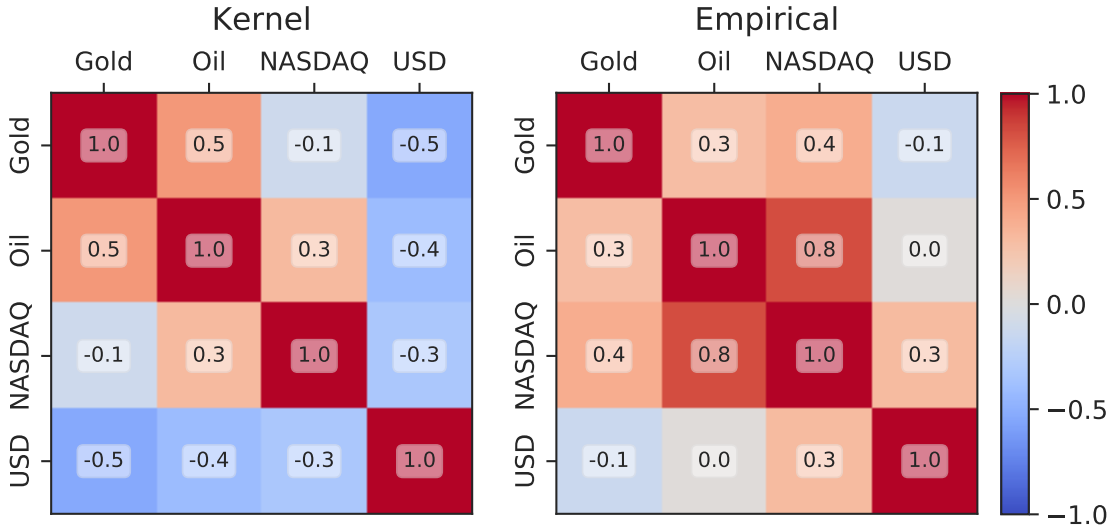

Figure 4.8: Cross-correlation matrix of the GONU data set (with missing data) among the channels of the trained MOSM by evaluating the (normalised) kernel (eq. 2.12) at $\tau = 0$ (left) and the empirical cross-correlation of the full data set (right). The off-diagonal elements show how much two currencies are aligned or anti-aligned, or whether they are unaligned and have negligible correlation.

Our trained MOSM kernel is recovering the more significant dependencies such as the oil and gold correlation and the oil and NASDAQ correlation. In Fig. 4.7 these curves follow similar behaviour, especially for oil and the NASDAQ this is apparent. The USD is found to correlate more negatively with the other channels, as well as gold and the NASDAQ. It should be noted that the MOSM finds correlations by minimising the negative log-likelihood (NLL), where if three channels correlate, the model could find correlation between the first and second, and between the second and third channels, but not necessarily between the

first and third, explaining the discrepancies between kernel and empirical cross-correlations. Furthermore, the MOSM only uses part of the data, and depending on the number of parameters and training it may not find all correlations. Table 4.5 shows error values of the test set comparing different models against the MOSM.

	Gold, Oil, NASDAQ, USD index	
Model	nMAE (10^{-2})	nRMSE (10^{-2})
SM-IGP	2.817 ± 0.000	5.071 ± 0.000
SM-LMC	2.5 ± 0.4	3.4 ± 0.6
CSM	1.88 ± 0.02	$\mathbf{2.44 \pm 0.06}$
MOSM	$\mathbf{1.8 \pm 0.1}$	2.6 ± 0.4

Table 4.5: Performance indices for the GONU and exchange rate experiments using the normalised mean absolute error (nMAE) and normalised root mean square error (nRMSE) on the test data and averaged over five test trials. Both are normalised by division of the mean.

4.6.2 Exchange Rates

Much like the GONU data set, the movement of exchange rates among large currencies is due to international market changes and national macro economic factors. Exchange rates are heavily influenced by inflation and interest rates, trade and economic performance. We chose ten exchange rates against the USD, namely the AUD, CAD, CHF, EUR, GBP, HKD, JPY, KRW, MXN, and NZD using a daily granularity with data ranging from 2017-01-01 to 2017-12-31. For all the channels, 30% of the data points have been removed randomly. All channels have the last 40 days removed except for EUR, JPY, and AUD. The EUR, JPY, and AUD thus act as reference channels to predict the other currency exchanges. For some channels an additional range has been removed to simulate missing data. Overall, we used 1535 training points and 955 test points, where each trial took roughly 60 minutes per trial for the MOSM. Table 4.6 shows error values of the test set comparing different models against the MOSM.

Fig. 4.9 shows the currency exchange data set with a fit of the MOSM kernel. We see that the predicted posterior means at the removed tails follow the data quite closely. A possible reason why one channel can recover missing data better while other channels have difficulty doing so, lies in the fact that a strongly correlating channel is needed to impute the data. Notice that since the MOSM is a covariance-driven model, the EUR, JPY, and AUD channels can be used to reconstruct the other channels.

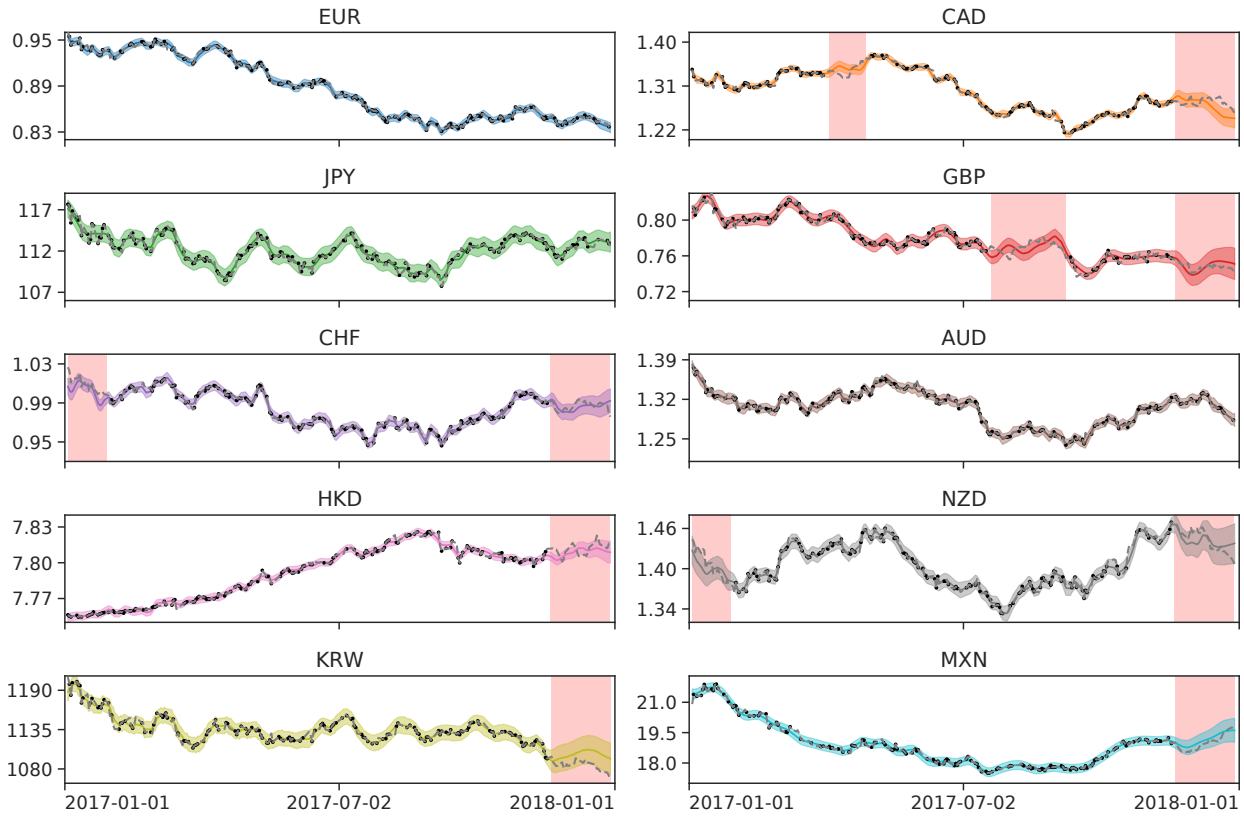

Figure 4.9: Ten currency exchange rates with respect to the USD fitted using the MOSM kernel. Training points are shown in black, ground truth in dashed grey, the coloured lines are the posterior means and the coloured shadows are the 95% confidence intervals. The red shaded areas mark the data imputation ranges.

Fig. 4.10 shows how much the channels correlate among each other under the trained MOSM kernel. Among the EUR, GBP, and CHF channels we see a strong positive correlation which is highly likely as the EU is the major trading partner for the GBP and CHF. Furthermore, we see that the HKD correlates negatively with the EUR, JPY, and AUD as the AUD and JPY correlate positively. The correlation between AUD and NZD is hardly surprising as these markets usually move quite similarly due to the geographic constraints of New Zealand.

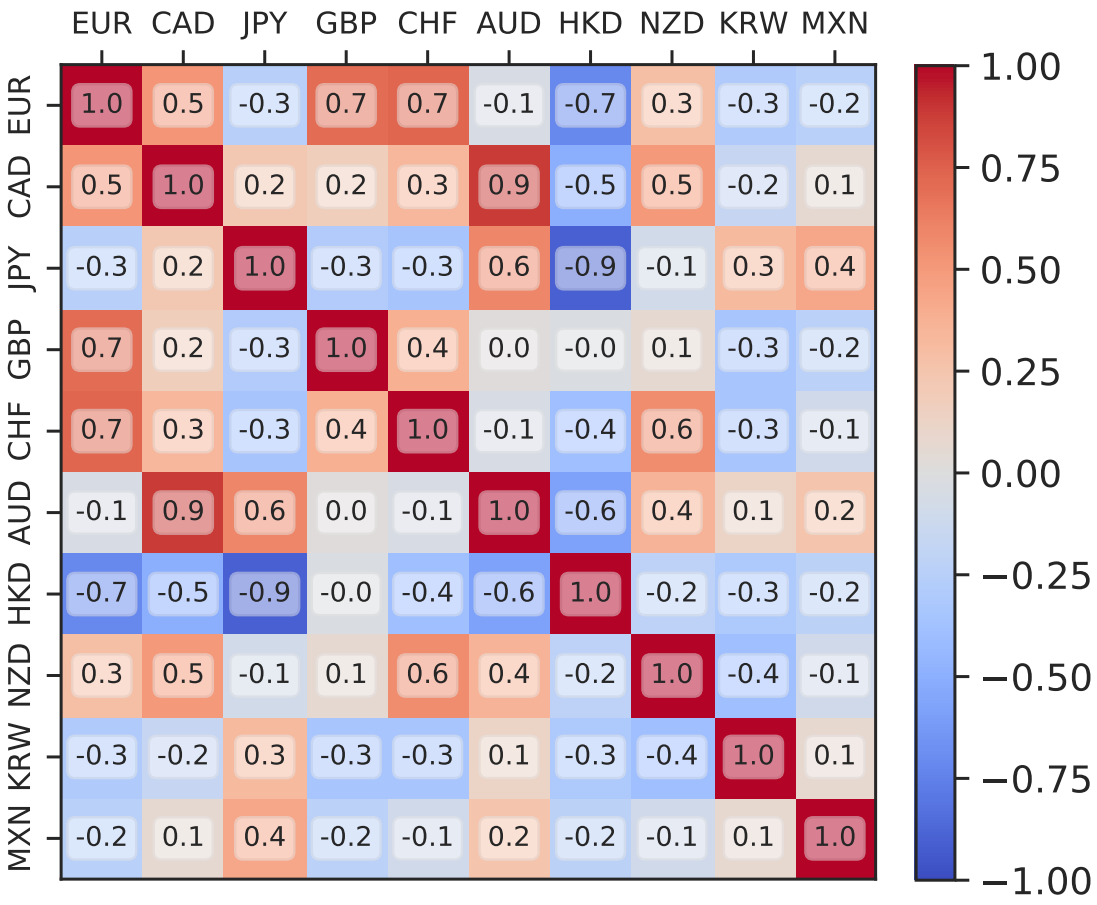

Figure 4.10: Cross-correlation between the ten currency exchange channels using the MOSM by evaluating the kernel (eq. 2.12) at $\tau = 0$ and normalising with the sum of the weights of each channel.

	Gold, Oil, NASDAQ, USD index		Currency exchange rates	
Model	nMAE (10^{-2})	nRMSE (10^{-2})	nMAE (10^{-3})	nRMSE (10^{-3})
SM-IGP	2.817 ± 0.000	5.071 ± 0.000	5.478 ± 0.000	7.481 ± 0.000
SM-LMC	2.5 ± 0.4	3.4 ± 0.6	6.6 ± 0.5	8.9 ± 0.6
CSM	1.88 ± 0.02	$\mathbf{2.44 \pm 0.06}$	8 ± 1	10 ± 2
MOSM	$\mathbf{1.8 \pm 0.1}$	2.6 ± 0.4	$\mathbf{4.8 \pm 0.3}$	$\mathbf{6.5 \pm 0.4}$

Table 4.6: Performance indices for the GONU and exchange rate experiments using the normalised mean absolute error (nMAE) and normalised root mean square error (nRMSE) on the test data and averaged over five test trials. Both are normalised by division of the mean.

Conclusions

We have revisited a framework for constructing flexible kernels for multi-output Gaussian processes (MOGP), by formulating a family of Hermitian functions which serve as the spectral representation of the kernel, then by taking the inverse Fourier transform, the multi-output kernel is formed, and also reinterpreted the framework as the convolution of latent white-noise processes with a filter, where the convolution between filters defines the kernel. This framework has been used with squared exponential functions as bases for the Hermitian family, resulting in the Multi-output spectral mixture (MOSM) kernel, where this covariance function can recover previous approaches to MOGP.

We also have expanded on the MOSM framework, tackling four main issues of the MOSM kernel: scalability for large datasets, difficulty in training due to sensibility to initial points, negative transfer of knowledge which can occur when the outputs are not correlated, and lack of user-friendly implementation available of current and previous models.

The main contributions of this thesis include: (i) addressing scalability of the model for large datasets by formulating sound inducing variables which can be used in variational sparse approaches, (ii) designing methods for finding an initial point in optimisation for spectral kernels, utilising the available data, (iii) by restricting certain components of the kernel to only influence specific outputs, where regularising using priors can mitigate the negative transfer of knowledge, which can occur when the outputs are not correlated, and (iv) implementing a toolkit containing the proposed MOSM, previous approaches which are particular cases of MOSM, and the aforementioned extension of the kernel.

To show that the MOSM kernel and the proposed variations constitute a robust approach to MOGP, we have validated the approach using real-world data, considering finance time series to assert the flexibility and predictive capabilities of the kernel, where MOSM outperformed previous approaches in MOGP. The integration of the kernel with current sparse approximation frameworks has been demonstrated utilising a robot-arm dataset with near 44000 training points, where the training only takes 15 minutes. The proposed kernel initialisations have been tested utilising a known periodic dataset on a single channel setup, where the proposed spectral initialisation using Bayesian non-parametric spectral estimation (BNSE) yielded the best results, however, these methods have shortcomings, wherein the case of multiple input dimension or a dataset with a non-pronounced periodic component the initialisation fails to deliver consistent results. Lastly, the regularising extension for mitigating negative transfer, restricted MOSM (R-MOSM), have been validated considering a synthetic dataset where the negative transfer can be controlled. Additional to this,

we have constructed the Multi-Output Gaussian Process Toolkit (MOGPTK), encapsulating the proposed-and-previous approaches to MOGP.

The proposed spectral initialisations constitute a robust method to find initial estimate of the kernel parameters, prior to optimisation, outperforming previous initialisation methods, in a dataset with a strong periodic component. Although this method can only be applied to kernels with known Fourier transform, this include commonly used kernels, such as squared exponential and Matérn, and specially the spectral mixture, where the spectral means constitute a sensible parameter in the optimisation. This method have been expanded to MOGP kernel, in particular, MOSM benefits from this due to the high number of kernel parameters, where the initial estimates helps the optimisation process.

A limitation of the proposed spectral initialisation, is that it yields insufficiency results when working with high number of input dimensions, where estimating the PSD for each input dimension is not enough to incorporate the information of the whole process.

In regard of R-MOSM, it is able to mitigate the negative transfer, when uncorrelated channels are present, by restricting certain components to only affect a limited number of channels. The R-MOSM outperforms classic MOSM in a setting where all channels are uncorrelated, and only the performance in a single channel is considered. Moreover, when increasing the number of correlated channels both methods yield similar results, noting that R-MOSM is able perform similar to MOSM when there is no risk of negative transfer. However, R-MOSM is not able to fully mitigate negative transfer, with a single-channel GP outperforming the R-MOSM, when all channels are uncorrelated, and focusing on the error of a single channel.

Regarding the proposed inducing variables, LIV has been incorporated in existing sparse approximation methods, enabling the MOSM kernel to be used in contexts with large quantities of data. The proposed LIV escalate better than classic inducing variables with the number of channels, with the trade-off of a high dependency of the number of components. As each inducing variable of the LIV is in the latent process, each inducing variable affects all channels, lowering the number of inducing variables necessary to represent the whole process, if the channels are highly correlated, but otherwise increasing the number necessary if the channels present low correlation.

The introduced MOGPTK have been successful in incorporating the MOSM kernel and the proposed extension, alongside previous models. The MOGPTK was used to perform all validation experiments, validating the use in real-world scenarios.

Future Directions

Future work includes the application of the proposed framework over other different base functions rather than the squared exponential, where and interesting option is the mixture of rectangular functions, which defines a limited-band spectrum, and will yield the recently proposed sinc kernel. [45].

The proposed framework for the MOSM kernel and its variants are expressive, simplifying

the user design by utilising the expressive kernel instead of relying on expert knowledge, however, there is still the issue of choosing the number of components Q, recent work [46] proposed the use of a non-parametric prior such as the Indian buffet process to model the number of components employing simple kernels, future work includes considering other non-parametric prior over the number of components of the MOSM kernel, such as Dirichlet processes or Determinantal point processes.

Concerning the proposed initialisation methods, an improvement on this would be to estimate the delay using the lag which produces the higher cross-correlation between a pair of channels, this is proposed as future work. An extension of this is not only to estimate the PSD of each channel for each input dimension, but estimate as well the cross-spectral densities to incorporate across-channel information before training. In the case with multiple input dimensions, the use of multivariate spectral estimation methods could be used instead of estimating the spectrum for each input dimension.

Regarding the proposed inducing variables, future improvements can be made using the spectral representation of the MOSM kernel, by designing inducing variables in the spectral domain of the latent processes [27], but instead of considering the spectrum of the output process, it uses the spectrum of the latent processes, benefiting from the independence of the latent factors.

Recent work [47] utilises an eigenfunction approximation of the covariance function utilising the spectral density for single output GP, this could be expanded for multi-output GP, where the MOSM kernel benefits as the spectral density is known. Another interesting line of work is at the variational Fourier features [48], which defines inducing variables by utilising an RKHS inner product between the process and a truncated Fourier base, this work was developed for the Matern kernel, as said inner product can be obtained in closed form, this could be expanded to arbitrary spectral kernels by employing a theorem which characterise the inner product of RKHS whose reproducing kernels have a closed-form Fourier transform [49] [50], this result states the following theorem,

Theorem 4.1 *Let k be a shift-invariant kernel on $\mathcal{X} = \mathbb{R}^{\mathrm{d}}$ such that $k(x, x') := k(x - x')$ for $k \in C(\mathbb{R}^{\mathrm{d}}) \cap L_1(\mathbb{R}^{\mathrm{d}})$. Consequently, the RKHS $\mathcal{H}_k$ of the kernel k will be given by,*

$$\mathcal{H}_k = \left\{ f \in C(\mathbb{R}^{\mathrm{d}}) \cap L_2(\mathbb{R}^{\mathrm{d}}) : ||f||^2_{\mathcal{H}_k} = \frac{1}{(2\pi)^{\mathrm{d}/2}} \int \frac{|F(\omega)|^2}{S(\omega)} \, \mathrm{d}\omega < \infty \right\}, \tag{4.9}$$

and the inner product of said RKHS will be given by,

$$< f, g >_{\mathcal{H}_k} = \frac{1}{(2\pi)^{\mathrm{d}/2}} \int \frac{F(\omega)\overline{G(\omega)}}{S(\omega)} \mathrm{d}\omega, \quad f, g \in \mathcal{H}_k, \tag{4.10}$$

where $F(\omega)$ and $G(\omega)$ correspond to the spectrum of f and g respectively, and $S(\omega)$ is the Fourier transform of the kernel.

This result could be used to expand the variational Fourier feature framework, allowing design of new inducing variables for spectral kernels.

Bibliography

[1] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[2] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for Vector-Valued Functions: A Review," *Foundations and Trends in Machine Learning*, 2012.

[3] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," 2013.

[4] G. Parra and F. Tobar, "Spectral mixture kernels for multi-output Gaussian processes," in *NeurIPS 30*, 2017.

[5] T. de Wolff, A. Cuevas, and F. Tobar, "MOGPTK: The Multi-Output Gaussian Process Toolkit," *accepted in Neurocomputing*, 2020.

[6] S. Bochner, *Lectures on Fourier Integrals: With an Author's Supplement on Monotonic Functions, Stieltjes Integrals and Harmonic Analysis; Translated by Morris Tenenbaum and Harry Pollard*. 1959.

[7] P. Goovaerts, *Geostatistics for Natural Resources Evaluation*. Oxford University Press, 1997.

[8] A. Wilson, *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[9] M. Alvarez and N. Lawrence, "Sparse convolved Gaussian processes for multi-output regression," in *NeurIPS 21*, 2009.

[10] K. Ulrich, D. Carlson, K. Dzirasa, and L. Carin, "GP kernels for cross-spectrum analysis," in *NeurIPS 28*, 2015.

[11] A. Yaglom, *Correlation theory of stationary and related random functions: Supplementary notes and references*. Springer Science & Business Media, 2012.

[12] C. Chatfield, "Time series analysis: an introduction," 1989.

[13] J. Quiñonero-Candela and C. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, 2005.

[14] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Artificial Intelligence and Statistics*, 2009.

[15] A. Matthews, J. Hensman, R. Turner, and Z. Ghahramani, "On sparse variational methods and the Kullback-Leibler divergence between stochastic processes," *Journal of Machine Learning Research*, 2016.

[16] M.Álvarez, D. Luengo, M. Titsias, and N. Lawrence, "Efficient multioutput Gaussian processes through variational inducing kernels," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[17] J. Hensman, N. Fusi, and N. Lawrence, "Gaussian processes for big data," in *Uncertainty in Artificial Intelligence*, 2013.

[18] D. B. A. Kucukelbir and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, 2017.

[19] J. Hensman, A. Matthews, M. Filippone, and Z. Ghahramani, "MCMC for variationally sparse Gaussian processes," in *Advances in Neural Information Processing Systems*, 2015.

[20] T. Sebastian, "Is learning the n-th thing any easier than learning the first?," in *Advances in Neural Information Processing Systems 8*, 1996.

[21] E. Bonilla, K. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Advances in neural information processing systems*, 2008.

[22] H. Liu, J. Cai, and Y. Ong, "Remarks on multi-output Gaussian process regression," *Knowledge-Based Systems*, 2018.

[23] C. Williams, S. Klanke, S. Vijayakumar, and K. Chai, "Multi-task Gaussian process learning of robot inverse dynamics," in *Advances in Neural Information Processing Systems*, 2009.

[24] T. de Wolff, A. Cuevas, and F. Tobar, "Gaussian process imputation of multiple financial series," in *ICASSP*, IEEE, 2020.

[25] Y. W. Teh, M. W. Seeger, and M. I. Jordan, "Semiparametric latent factor models.," in *AISTATS*, Society for Artificial Intelligence and Statistics, 2005.

[26] F. Tobar., T. D. Bui, and R. E. Turner, "Learning stationary time series using Gaussian processes with nonparametric kernels," in *Advances in Neural Information Processing Systems 28*, 2015.

[27] M. Lázaro-Gredilla and A. Figueiras-Vidal, "Inter-domain Gaussian processes for sparse inference using inducing features," in *Advances in Neural Information Processing Systems*, 2009.

[28] F. Tobar, "Bayesian nonparametric spectral estimation," *NeurIPS 31*, 2018.

[29] A. Schuster, "On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena," *Terrestrial Magnetism*, 1898.

[30] P. Welch, "The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on audio and electroacoustics*, 1967.

[31] N. R. R. Lomb, "Least-squares frequency analysis of unequally spaced data," *Astrophysics and space science*, 1976.

[32] D. J. J. Scargle, "Studies in astronomical time series analysis. ii-statistical aspects of spectral analysis of unevenly spaced data," *The Astrophysical Journal*, 1982.

[33] G. Leen, J. Peltonen, and S. Kaski, "Focused multi-task learning in a Gaussian process framework," *Machine Learning*, 2012.

[34] R. Kontar, G. Raskutti, and S. Zhou, "Minimizing Negative Transfer of Knowledge in Multivariate Gaussian Processes: A Scalable and Regularized Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[35] V. Tresp, "A bayesian committee machine," *Neural computation*, 2000.

[36] A. Matthews *et al.*, " GPflow: A Gaussian process library using TensorFlow," *Journal of Machine Learning Research*, 2017.

[37] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 2016.

[38] R. Keeling, S. Piper, A. Bollenbacher, and J. Walker, "Atmospheric CO2 records from sites in the SIO air sampling network." In Trends: A Compendium of Data on Global Change, 2009.

[39] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space," in *Proceedings International Conference on Machine Learning (ICML)*, 2000.

[40] T. V. Nguyen and E. V. Bonilla, "Collaborative multi-output Gaussian processes," in *UAI*, 2014.

[41] J. Zhao and S. Sun, "Variational dependent multi-output Gaussian process dynamical systems," *The Journal of Machine Learning Research*, 2016.

[42] "Brent oil price." `https://www.eia.gov/dnav/pet/hist/RBRTEd.htm`. Accessed: 2019-09-01.

[43] "NASDAQ price." `https://finance.yahoo.com/quote/%5EIXIC/history?p=%5EIXIC`. Accessed: 2019-09-01.

[44] "Trade weighted USD-index against the currencies of a broad group of trading partners

from January 1995 till August 2019." https://fred.stlouisfed.org/series/TWEXB. Accessed: 2019-09-01.

[45] F. Tobar, "Band-limited Gaussian processes: The sinc kernel," in *Advances in Neural Information Processing Systems*, 2019.

[46] A. Tong and J. Choi, "Discovering latent covariance structures for multiple time series," in *International Conference on Machine Learning*, PMLR, 2019.

[47] A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank Gaussian process regression," *Statistics and Computing*, 2020.

[48] J. Hensman, N. Durrande, and A. Solin, "Variational fourier features for Gaussian processes," *The Journal of Machine Learning Research*, 2017.

[49] H. Wendland, *Scattered Data Approximation*. Cambridge University Press, 2004.

[50] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur, "Gaussian processes and kernel methods: A review on connections and equivalences," *arXiv preprint arXiv:1807.02582*, 2018.

Appendix A.

In this appendix we derive the expression from Section 2.4, eq. (2.14), where the MOSM kernel can be obtained from the convolution of a white noise process and a filter $h_{\mathrm{i}}(x)$.

Proof.

$$
\begin{aligned}
h_{\mathrm{i}}(x) &= \mathcal{F}^{-1}\left\{R_{\mathrm{i}}(\omega)\right\} \\
&= \mathcal{F}^{-1}\left\{a_{\mathrm{i}} \exp\left(-\frac{1}{4}(w-\mu_{\mathrm{i}})^{\top}\Sigma_{\mathrm{i}}^{-1}(w-\mu_{\mathrm{i}})\right)\exp\left(-\iota(\theta_{\mathrm{i}}^{\top}w+\phi_{\mathrm{i}})\right)\right\} \\
&= a_{\mathrm{i}}\mathcal{F}^{-1}\left\{\exp\left(-\frac{1}{4}(w-\mu_{\mathrm{i}})^{\top}\Sigma_{\mathrm{i}}^{-1}(w-\mu_{\mathrm{i}})\right)\right\} \star \mathcal{F}^{-1}\left\{\exp\left(-\iota\theta_{\mathrm{i}}^{\top}w\right)\exp\left(-\iota\phi_{\mathrm{i}}\right)\right\} \\
&= a_{\mathrm{i}}\left\{|\Sigma_{\mathrm{i}}|^{1/2}(2\pi)^{n/2}\exp(-x^{\top}\Sigma_{\mathrm{i}}x)\exp(\iota x^{\top}\mu_{\mathrm{i}})\right\} \star \left\{\delta(x-\theta_{\mathrm{i}})\exp(-\iota\phi_{\mathrm{i}})\right\} \\
&= a_{\mathrm{i}}|\Sigma_{\mathrm{i}}|^{1/2}(2\pi)^{n/2}\exp(-\iota\phi_{\mathrm{i}})\left\{\delta(x-\theta_{\mathrm{i}})\star \exp(-x^{\top}\Sigma_{\mathrm{i}}x)\exp(\iota x^{\top}\mu_{\mathrm{i}})\right\} \\
&= a_{\mathrm{i}}|\Sigma_{\mathrm{i}}|^{1/2}(2\pi)^{n/2}\exp(-\iota\phi_{\mathrm{i}})\exp\left(-(x+\theta_{\mathrm{i}})^{\top}\Sigma_{\mathrm{i}}(x+\theta_{\mathrm{i}})\right)\exp\left(\iota(x-\theta_{\mathrm{i}})^{\top}\mu_{\mathrm{i}}\right) \\
&= a_{\mathrm{i}}|\Sigma_{\mathrm{i}}|^{1/2}(2\pi)^{n/2}\exp\left(-(x+\theta_{\mathrm{i}})^{\top}\Sigma_{\mathrm{i}}(x+\theta_{\mathrm{i}})\right)\exp\left(\iota\left((x-\theta_{\mathrm{i}})^{\top}\mu_{\mathrm{i}}-\phi_{\mathrm{i}}\right)\right).
\end{aligned}
$$

□

Appendix B.

	L-BFGS-B SMSE	CG SMSE	Adam SMSE
Random	0.240 ± 0.097	0.310 ± 0.002	0.238 ± 0.015
IPS	2.447 ± 2.628	0.733 ± 1.479	0.649 ± 1.496
GMM	0.310 ± 0.001	0.299 ± 0.022	0.288 ± 0.000
LS	0.162 ± 0.000	0.200 ± 0.000	0.120 ± 0.000
LS+ϵ	0.246 ± 0.140	0.142 ± 0.095	0.177 ± 0.101
BNSE	0.071 ± 0.000	0.008 ± 0.000	0.185 ± 0.000
BNSE+ϵ	0.322 ± 0.332	0.206 ± 0.110	0.142 ± 0.062

Table 7: SMSE for different initialisations and optimisers in GP regression using SM for Mauna Loa dataset, results averaged over 10 trials.

Appendix C.

Tables of experiment of negative transfer of knowledge, used in Fig.4.1 and Fig.4.2.

	RMSE			
	SM	MOSM	R-MOSM	R-MOSM-P
0-Correlated channels	0.538	1.041 ± 0.149	0.701 ± 0.256	0.643 ± 0.119
1-Correlated channels	-	0.591 ± 0.352	0.598 ± 0.146	0.640 ± 0.065
2-Correlated channels	-	0.566 ± 0.195	0.532 ± 0.101	0.594 ± 0.208
3-Correlated channels	-	0.390 ± 0.258	0.436 ± 0.189	0.537 ± 0.173
4-Correlated channels	-	0.367 ± 0.337	0.344 ± 0.170	0.261 ± 0.095

Table 8: RMSE for primary channel at different number of correlated channels, results averaged over 5 trials.

	SMSE			
	SM	MOSM	R-MOSM	R-MOSM-P
0-Correlated channels	0.319	1.879 ± 1.194	0.521 ± 0.317	0.430 ± 0.146
1-Correlated channels	-	0.472 ± 0.434	0.368 ± 0.161	0.414 ± 0.060
2-Correlated channels	-	0.257 ± 0.142	0.204 ± 0.056	0.256 ± 0.113
3-Correlated channels	-	0.172 ± 0.216	0.219 ± 0.225	0.271 ± 0.120
4-Correlated channels	-	0.284 ± 0.495	0.119 ± 0.108	0.072 ± 0.062

Table 9: SMSE for primary channel at different number of correlated channels, results averaged over 5 trials.

	NLPD			
	SM	MOSM	R-MOSM	R-MOSM-P
0-Correlated channels	0.643	1.633 ± 0.334	1.309 ± 0.819	1.055 ± 0.282
1-Correlated channels	-	0.823 ± 0.700	1.026 ± 0.473	1.348 ± 0.432
3-Correlated channels	-	1.368 ± 1.309	1.211 ± 0.824	2.235 ± 2.115
2-Correlated channels	-	0.534 ± 0.741	0.582 ± 0.481	1.073 ± 0.798
4-Correlated channels	-	0.320 ± 0.663	0.338 ± 0.427	0.157 ± 0.258

Table 10: NLPD for primary channel at different number of correlated channels, results averaged over 5 trials.